大数据分析：R基础及应用

走进R，走进大数据时代数据分析的潮流尖端，掌握R语言，
熟悉大数据的基础概念和R与Hadoop结合进行大数据的处理分析。

深圳国泰安教育技术股份有限公司
中科院深圳先进技术研究院-国泰安金融大数据研究中心 编著

深圳国泰安教育技术股份有限公司是国家科技部重点支持的国家级高新技术企业和国家重点软件企业，是知名的中国经济金融数据权威提供商、中国教学研究服务行业领导者，致力于为教育业和金融业提供一流产品、增值服务及软硬件整体解决方案。

清華大學出版社

大数据分析：R 基础及应用

深圳国泰安教育技术股份有限公司
中科院深圳先进技术研究院—国泰安金融大数据研究中心 编著

清华大学出版社
北 京

内 容 简 介

在大数据时代，R以其强大的数据分析挖掘、可视化绘图等功能，越来越受到社会各个领域的青睐。现在，R的计算引擎、性能、程序包都得到了提升，其中R与大数据分析平台Hadoop的结合，实现了R对大数据的分析式处理分析。这些不仅大大扩展了R的应用，也扩大了R在各行业的需求。

为了更好地适应新形势，掌握大数据分析处理的相关知识是很有必要的。本书从理论基础、方法、实证三方面详细地阐释了R和RHadoop的相关理论、技术以及应用，使读者了解大数据的基础概念，掌握R以及Rhadoop大数据分析技术。本书不仅适合高等院校的各相关专业的本专科生、研究生，也适合零编程基础的科研人员以及对大数据分析技术感兴趣的人士阅读。本书在内容的选择和结构的安排上进行了深入的思考，使得不论是R或RHadoop的初学者还是具备一定相关专业知识的人员都能从本书中得到一定的收获或启发。

图书在版编目(CIP)数据

大数据分析：R基础及应用/深圳国泰安教育技术股份有限公司，中科院深圳先进技术研究院－国泰安金融大数据研究中心编著. --北京：清华大学出版社，2016

ISBN 978-7-302-42863-3

Ⅰ. ①大… Ⅱ. ①深… ②中… Ⅲ. ①数据处理软件 Ⅳ. ①TP274

中国版本图书馆 CIP 数据核字(2016)第 024162 号

责任编辑：彭 欣
封面设计：
责任校对：王荣静
责任印制：

出版发行：清华大学出版社
网 址：http://www.tup.com.cn，http://www.wqbook.com
地 址：北京清华大学学研大厦A座 **邮 编**：100084
社 总 机：010-62770175 **邮 购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经 销：全国新华书店
开 本：185mm×260mm **印 张**：12.5 **字 数**：300千字
版 次：2016年3月第1版 **印 次**：2016年3月第1次印刷
印 数：1～ 000
定 价：49.00元

产品编号：068005-01

编　　委

编撰单位：

深圳国泰安教育技术股份有限公司

中科院深圳先进技术研究院——国泰安金融大数据研究中心

主编：

陈工孟，深圳国泰安教育技术股份有限公司董事长，上海交通大学教授、博导

须成忠，中科院深圳先进技术研究院数字所所长，云计算研究中心主任、教授、博导

执行主编：

贾淑芹，深圳国泰安教育技术股份有限公司大数据事业部群副总经理

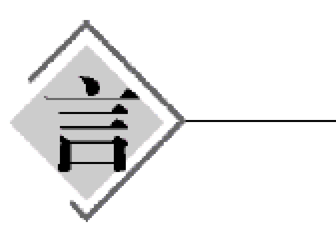
姜义平，中科院深圳先进技术研究院——国泰安金融大数据研究中心常务副主任、深圳国泰安教育技术股份有限公司大数据事业部群副总经理

凌宗平，深圳国泰安教育技术股份有限公司大数据事业部群副总经理、中国量化投资研究院助理院长

李晓龙，深圳国泰安教育技术股份有限公司大数据学术事业部总经理

编撰人员：

李敏、朱清、陈文慧、张金秀、祝雨露、阮志锋、刘瑶

FOREWROD 引 

大数据时代，R被拉到了潮流尖端，作为免费的开源软件，随着加入的人数增多，R的计算引擎、性能、各种程序包都得到了改进和升级，其中R和Hadoop的结合RHadoop实现了大规模数据的分布式处理分析，RHive包将R语言与Hive连接，可以通过R快速访问存储在Hive的大数据集，这一切让R获得了新生。为了更好地适应新形势，国泰安联合中科院先进院于2014年10月成立了金融大数据研究中心。鉴于此，国泰安大数据事业部群组织专家学者推出了《大数据分析：R基础及应用》一书，该书具有以下几个方面的特色。

1. 实训性强

目前，市面上流通的R语言经济金融建模系列教材不胜枚举。本书的特色在于选取特定的专题来解决一些实际问题，让读者学习如何使用R语言进行实证建模。同时，本书也给出了一些非常有价值的总结和后续思考，以供读者研究。

2. 编排体系合理

整个的结构按照"大数据简介→R语言基础知识→数据分析功能→专题实证研究→RHadoop案例分析"这样的思路组织全书，既方便读者(特别是初学者)在了解大数据概念和技术的基础上学习R软件的操作和简单编程，也帮助他们快速地用R语言建立模型，并作出分析和结果论证，有大量的案例可作参考。

3. 考虑不同群体的阅读偏好和水平

本书涉及面广，在专题实证研究部分涵盖了多个领域，包括金融时间序列建模专题、动态面板数据专题、大数据时代数据挖掘专题、机器学习专题和信息可视化专题，充分展示了当前该领域的需求和R的强大优势。

本书适合没有编程基础的科研人员及大数据分析人员使用。从事经管类的学术研究往往都需要建模及数据作为支撑。本书分为三大部分进行介绍，即理论基础+方法+实证。理论基础分为两个章节，主要介绍大数据的基础知识和相关技术。方法部分分为4个章节，其中第3章主要对R语言进行简单的介绍，第4章是R语言的操作讲解；第5章将介绍R语言一大特色——可视化图表及相关统计分析的R语言实现；第6章将对R语言数据分析处理进行一个简单介绍。实证部分包括专题实证研究和RHadoop案例分析，其中专题实证研究介绍4个专题，给出不同的实际案例，循序渐进地讲解如何利用R语言进行实证建模，包括时间序列模型、动态面板数据模型、数据挖掘及信息可视化。这些模型既涵盖了

理论的指导，又附有程序的说明及结果的验证，同时还包括对模型进一步的延伸与思考。RHadoop 案例分析部分介绍在 RHadoop 环境下 R 的基本操作及 8 个案例，包括回归分析、logistic 分析、判别分析、聚类分析、主成分分析、因子分析、商品推荐算法及差异分析，针对不同的分析方法介绍算法的原理和 RMapReduce 编程实现。

本书编写组希望《大数据分析：R 基础及应用》一书可以对广大读者有所帮助，相信读者能收获以下几点：

1. 掌握大数据的基础概念和 R 处理大数据的机制，并深入地了解 R 语言，能够掌握 R 编程的基本技能，程序注释非常清楚，易学易懂。

2. 熟练掌握从建模到利用 R 语言对数据进行实证的整个过程。

3. 可以学习金融时间序列建模，数据挖掘等领域的一些比较经典和前沿的热门模型。

4. 能够学习到不同学科之间的交叉应用，包括统计学与金融，数学与金融等一系列知识。

5. 熟悉 RHadoop 环境，掌握 RMapReduce 编程，实现在 RHadoop 环境下进行大数据分析。

限于编者的能力和时间，本书难免存在纰漏或不足之处，欢迎读者批评指正。

深圳国泰安教育技术股份有限公司

CONTENTS 目 录



第一部分　大数据简介

第二部分 R 语言

第三部分　专题实证研究

PART 1

第一部分

大数据简介

CHAPTER 1

第1章 大数据概述

大数据时代早已到来,《大数据时代》的作者维克托·迈尔·舍恩伯格说,世界的本质就是数据,大数据将开始一次重大的时代转型。其实早在 1980 年,美国著名未来学者托夫勒便在《第三次浪潮》一书中提出“数据就是财富”,将大数据热情地赞颂为“第三次浪潮的华彩乐章”。作为云计算领域的重要延伸,大数据正在引领信息革命进入新的时代。2001 年,全球最具权威的 IT 研究与顾问咨询公司 Gartner 提出大数据面临 4 个 V 的挑战;《自然》杂志(2008 年)推出《大数据》专刊,全方面介绍大数据问题;美国总统奥巴马(2012 年)将数据定义为“未来的新石油”。2013 年,Gartner 在一篇报告中指出,64%的受访企业都表示他们正在或是即将进行大数据工作。信息技术、计算机技术和互联网技术的迅速发展,使得人类社会各类数据呈现出爆炸性增长,对这些复杂大数据的有效管理,现已成为当前社会的热点问题。

1.1 大数据的概念

大数据(Big Data),或称为巨量资料,指的是所涉及的资料量规模巨大到无法通过目前主流软件工具,在合理时间内达到撷取、管理、处理并整理成为帮助企业经营决策目的的资讯。大数据一般指在 10TB(1TB=1024GB)规模以上的数据量,其基本特征可以用 4 个 V 来总结:数据规模大(Volume)、数据类别多(Variety)、数据处理速度快(Velocity)、价值密度低(Value)[①]。

然而,“大数据”的概念远不止大量的数据(TB)和处理大量数据的技术,或者所谓的“4 个 V”之类的简单概念,而是涵盖了人们在大规模数据的基础上可以做的事情,而这些事情在小规模数据的基础上是无法实现的。换句话说,大数据让我们以一种前所未有的方式,通过对海量数据进行分析,获得有巨大价值的产品和服务,或深刻的洞见,最终形成变革之力。

---

① http://www.cfern.org/wjgg/wjggDisplay.asp? Id=2353.

1.2 大数据的特征

大数据具有以下 4 个基本特征：数据规模大、数据类别多、数据处理速度快、价值密度低。

1. 数据规模大

大数据的基本属性是数据量巨大。目前，各个行业中的各个企业每天都会产生大量的数据，数据呈爆炸式的增长，数据量已从 TB 级别跃升到 PB 级别，甚至到了 EB 数量级。面对海量数据，传统的数据库系统处理能力已经难以应对，而且数据量仍在大规模增长，产生数据的来源也变得更加多样化。

2. 数据类别多

大数据除了传统的商业活动产生的数据外，还包括互联网上社交媒体产生的文本数据及时刻产生的传感器数据等。数据类型除了结构化数据外，还有半结构化和非结构化数据，如图片、网页、视频等，数据种类繁多。

3. 数据处理速度快

大数据和传统数据挖掘最显著的一个区别就是大数据要求处理速度快。面对如此大规模的数据，有效处理数据的效率也就牵系着企业的命运。对数据的实时处理、分析及反馈变得十分重要，创建实时数据已经成为一种趋势。

4. 价值密度低

价值密度往往与数据量成反比，在大量数据中有用的信息可能是非常少的，而且要有效地获取这些有用的信息也是比较困难的。比如，连续的监控产生大量的视频信息，而我们需要的数据可能就只有一两秒。针对大数据价值密度低这一特征，如何有效地挖掘出其中有用信息变得尤为重要。

1.3 大数据的产生

大数据的产生是计算机和网络通信技术被广泛运用的必然结果。互联网、移动互联网、物联网、云计算、社交网络等新一代信息技术的发展对大数据的产生起到了促进的作用。数据产生方式的变化表现为以下 4 个方面。

(1) 数据产生由企业内部向企业外部扩展。

(2) 数据产生由 Web1.0 向 Web2.0 扩展。

(3) 数据产生由互联网向移动互联网扩展。

(4) 数据产生由计算机或互联网(IT)向物联网(IOT)扩展。

这 4 个方面的变化让数据产生的源头成几何数增长，数据量也呈现出大幅度地快速增加。

1.4 大数据应用案例

大数据在各行业中有着大量的应用案例，比如金融行业中的信贷分析、银行风险分析及公司的交易分析等，医疗行业中的流行病学研究、病房的实时监控等，以及在亚马逊、淘宝

网、Facebook 等互联网企业中的应用等。下面给出一个典型的大数据应用案例——余额宝。

余额宝的问世改变了天弘基金由原来国内排名中下并且连年亏损的状态，使得它位居国内基金管理公司之首，世界排名 14。该公司将天弘增利宝货币基金从零开始发展到用户数量超过 1 亿元、资金规模达到 5742 亿元，超出了预计的 10 倍，成为世界第四大货币基金。

余额宝产生的背景是天弘基金欲借助最大电商阿里平台，在支付宝上向用户推销基金。阿里负责余额宝在支付宝端的建设，天弘基金负责与支付宝对接的直销和清算系统的建设。面对大规模的数据量，余额宝之前的系统已经不能满足需求，需要重建。余额宝的系统建设分为两期，然而随着数据量和交易量暴增，使得第一期系统仍无法负载日益增长的海量数据。于是进行了第二期系统的建设，阿里金融云提供了云计算服务，使得该系统的性能得到了相当大的提高，在很大程度上缩短了清算时间。在 2013 年 11 月 11 日的“双 11”活动中，余额宝完成了 1679 万笔赎回，1288 万笔申购的清算工作，成功为 639 万用户正确分配收益，当天处理了 61.25 亿元的消费赎回，119.97 亿元的转入申购，而系统只用了 46 分钟就将全部清算工作完成。

实际上，二期系统现已不是简单的直销和清算系统，它每天面对着 50 个数据库里海量用户和交易数据的暴涨。那么，这些数据的使用、价值最大化吸引了企业机构的眼球。对此，天弘基金选择了阿里云提供的 ODPS（开放数据处理服务）作为大数据平台，其中 ODPS 是阿里集团进行离线数据处理的平台，支撑了阿里金融、淘宝等多家 BU 的大数据业务。天弘基金将目标锁定在余额宝产生的海量数据分析上，以求把握上亿用户的理财需求及不同的风险接受能力，创造出更多更丰富的理财产品[①]。

① http://www.csdn.net/article/2014-05-26/2819939.

CHAPTER 2

第2章 大数据相关技术

大数据处理流程主要是指从海量数据中获取需要的信息并进行加工分析得到有用知识的输出过程。大数据处理流程的关键技术包括大数据存储和管理及大数据检索使用(包括数据挖掘和智能分析)。围绕大数据,一批新兴的数据存储、数据挖掘、数据处理与分析技术不断涌现,使得对海量数据的处理变得更加简便快速。大数据处理流程一般包括以下几个步骤:数据采集/清洗、数据存储、数据挖掘及数据呈现,如图 2.1 所示。

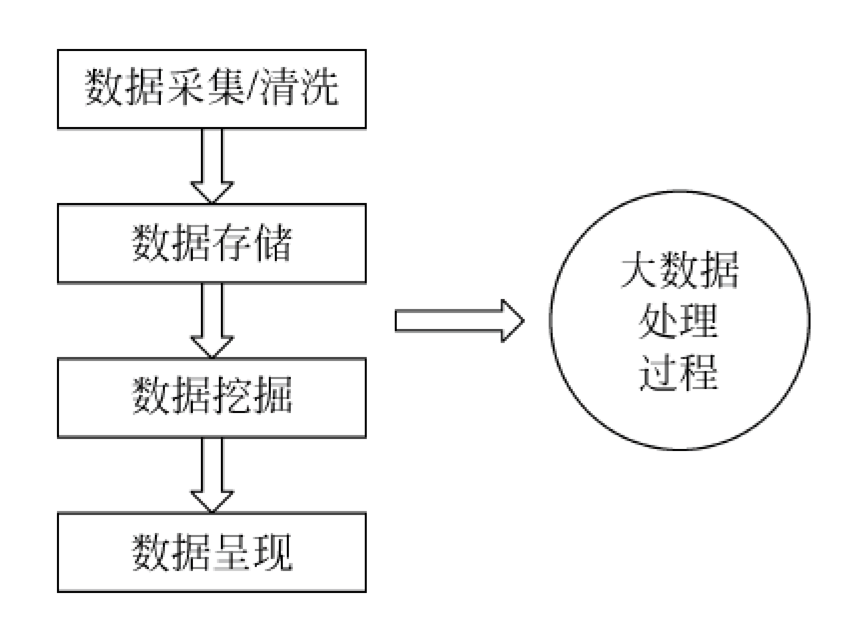


图 2.1　大数据处理流程

2.1　数据采集和准备

数据采集,即数据获取,是指从传感器或其他待测设备中获取信息的过程[①]。大数据采集包括对实时数据、非实时数据的采集,数据类型包括结构化、半结构化及非结构化数据。

大数据采集的方法有系统日志采集、数据库采集、网络数据采集等,采集的工具包括传感器、网络爬虫、移动基站及使用者自身产生的信息。

① http://baike.baidu.com/link?url=lnD8lmwKE4vGOneQhSBhNfFPNt7MfXl-sSyubVzcdYMN2Xsf9ylWBOLSLZt0YpVWInArgZunuSpSgv6G2bGrI_.

1. 传感器

传感器是一种检测装置，它采集数据的过程为：首先传感器感受被测量的信息，然后将其按一定规律变换成为电信号或其他形式的信息并输出。传感器是大规模数据的来源，比如，监控大型强子对撞机或四发动机大型喷气式客机需要成千上万的传感器通道，从而产生数百 TB 的数据。

2. 网络爬虫

网络爬虫是一种按照一定的规则，自动提取互联网网页信息的程序或脚本。互联网的数据形式多样，包括结构化的数据及图片、音频、视频等非结构化数据，对于这些海量数据，传统的获取方法已经不能满足需求，所以网络爬虫技术应运而生。网络爬虫可以定向地抓取用户所需的与某一特定主题相关的网页内容。

3. PON

日常通信过程中产生的海量信息。

4. 使用者自身产生的信息

随着微信、微博及邮件等的普及，使得它们拥有庞大的用户群。在人们使用这些软件的同时会产生巨大的信息，这些信息也是海量数据的重要来源。

在进行数据挖掘与分析前需要对数据进行一定的处理，即数据的准备。数据的准备是数据分析整个过程中的一个重要阶段，可以为后续的挖掘分析提供高质量的数据，从而保证了分析结果的有效性。数据准备包括数据的导入、数据的抽取、转换和装载等。数据导入指的是将外部数据导入到数据库或数据仓库中，关键是针对数据库的存储方式及具体的应用场景定义数据合适的模式。数据的抽取(Extract)是指将所需数据从源数据中抽取出来；数据的转换(Transform)是将获取的源数据按照一定的业务需求转换成所需要的形式，包括对数据的清洗和加工等操作；数据的装载(Load)指的是将经过转换后的数据装载到目的数据数据源中。ETL 过程包括对数据空值的处理、数据格式的规范化处理、数据的替换及正确性验证的处理等，是数据挖掘分析的基础。

2.2　分布式数据库

大数据包括结构化数据、半结构化数据及非结构化数据，大数据的存储与普通数据存储的差别主要表现在数量级别和能否存储索引非结构化数据上。对于声音、图片、视频等非结构化数据，传统的关系型数据库无法满足存储需求，因此非关系型数据库变得尤为重要。大数据处理系统将通过 NoSQL 来存储这些非结构化数据并对这些数据进行相关的检索。

NoSQL 数据库指的是非关系型的数据库。NoSQL 数据库主要面向 Web 应用，支持分布式存储，能够满足对数据库高并发读写需求、海量数据的高效存储需求、数据库高扩展性和高可用性的需求等。NoSQL 数据库可以分为以下三类：面向高性能读写的数据库、面向文档的数据库及面向分布式计算的数据库(比如 Cassandra 数据库)。NoSQL 具有自由灵活的数据模型，典型的 NoSQL 数据库是以键值(Key-Values)的形式存储数据的。

NoSQL 满足 CAP 理论、BASE 原则。CAP 指的是对于以下三个特性：一致性、可用性及分区容错性，分布式系统不能同时满足，最多只能满足三个特性中的两个。BASE 指的是 Basically Available、Soft state、Eventually consistent。Basically Available(基本可用)指的

是对于系统短时间内的不可用是可容忍的；Soft state（柔性状态）指的是系统有异步的情况存在，即在某个时期可以不同步；Eventually consistent（最终一致性）指的是只要最终的数据满足一致性即可，不要求时刻满足一致性。NoSQL 数据库的设计一般针对具体的应用，遵循以上两个原则，比较注重数据的读写效率、数据的容量和系统的可扩展性等。

目前普遍使用的关系型数据库采用的是关系型数据模型，对数据存储增加及一些需要满足的数据范式，有时需要强行修改对象数据，以满足关系型数据库管理系统的需要，而 NoSQL 数据库完全改变了传统的观念，通过改变某些数据范式的严格要求，获得灵活的扩展性、灵活的数据模型、能够有效处理大数据、降低管理和维护成本等众多优点。表 2.1 对 NoSQL 数据库与关系型数据库的原理、规模、模式等进行了一个对比分析。

表 2.1　NoSQL 和关系型数据库的简单比较

比较标准	RDBMS	NoSQL	备　　注
数据库原理	完全支持	部分支持	RDBMS 有数学模型支持，NoSQL 则没有
数据规模	大	超大	RDBMS 的性能会随着数据规模的增大而降低；NoSQL 可以通过添加更多设备以支持更大规模的数据
数据库模式	固定	灵活	使用 RDBMS 需要定义数据库模式，NoSQL 则不用
查询效率	快	简单查询非常高效、较复杂的查询性能有所下降	RDBMS 可以通过索引，能快速地响应记录查询（point query）和范围查询（range query）；NoSQL 没有索引，虽然 NoSQL 可以使用 MapReduce 加速查询速度，但仍然不如 RDBMS
一致性	强一致性	弱一致性	RDBMS 遵守 ACID 模型；NoSQL 遵守 BASE（Basically Available、Soft State、Eventually Consistent）模型
扩展性	一般	好	RDBMS 扩展困难；NoSQL 扩展简单
可用性	好	很好	随着数据规模的增大，RDBMS 为了保证严格的一致性，只能提供相对较弱的可用性；NoSQL 任何时候都能提供较高的可用性
标准化	是	否	RDBMS 已经标准化（SQL）；NoSQL 还没有行业标准
技术支持	高	低	RDBMS 经过几十年的发展，有很好的技术支持；NoSQL 在技术支持方面不如 RDBMS
可维护性	复杂	复杂	RDBMS 需要专门的数据库管理员（DBA）维护；NoSQL 数据库虽然没有 DBMS 复杂，但是也难以维护

随着互联网 Web 2.0 网站的兴起，传统的关系数据库在应付 Web 2.0 网站，特别是超大规模和高并发的 SNS 类型的 Web 2.0 纯动态网站已经显得力不从心，暴露了很多难以克服的问题，非关系型的数据库则由于其本身的特点得到了非常迅速的发展。

在信息技术融合应用的新时代，大数据就是像黄金一样的新型经济资产、像石油一样的重要战略资源。为满足大数据对处理和存储能力的无限需求，现今的计算机体系结构在数据存储方面要求具备庞大的水平扩展性（Horizontal Scalability，即要求满足能够连接多个软硬件的特性，这样可以将多个服务器从逻辑上看成一个实体），而 NoSQL 致力于改变这一现状。目前 Google 的 BigTable 和 Amazon 的 Dynamo 使用的就是 NoSQL 数据库。NoSQL 数据库根据数据的存储模型和特点分为很多种类，如列存储、文档存储、Key-Value 存储、图存储、对象存储、xml 存储等数据库。表 2.2 给出了几种典型的 NoSQL 数据库及

其性能优缺点。

表 2.2　典型的 NoSQL 数据库分类

NoSQL 数据库类型	代表性产品	性能	扩展性	灵活性	复杂性	优点	缺点
键/值数据库	Redis Riak	高	高	高	无	查询效率高	不能存储结构化信息
列式数据库	HBase Cassandra	高	高	一般	低	查询效率高	功能较少
文档数据库	CouchDB MongoDB	高	可变	高	低	数据结构灵活	查询效率较低
图形数据库	Neo4J OrientDB	可变	可变	高	高	支持复杂的图算法	只支持一定的数据规模

在过去的 10 年里，正如交易率发生了翻天覆地的增长一样，需要存储的数据量也发生了急剧的膨胀，这种现象被称为“数据的工业革命”。为了满足数据量增长的需要，RDBMS（关系型数据库管理系统）的容量也在日益增加，但是对于一些企业来说，随着交易率的增加，单一数据库需要管理的数据约束的数量也变得越来越让人无法忍受了。现在，大量的“大数据”可以通过 NoSQL 系统来处理，它们能够处理的数据量远远超出了最大型的 RDBMS 所能处理的极限，很好地弥补了关系数据在某些方面的不足。

2.3　分布式数据分析框架

对于海量数据处理，一般可以分成离线数据处理和流式数据处理两大类。在海量数据的计算中，Hadoop 无疑是开源分布式离线处理技术的一大主力，而 Storm 则提供了分布式流处理框架，让实时大数据处理得以实现。

2.3.1　Hadoop

Hadoop 是一个能够对大量数据进行分布式处理的软件框架，并且是以一种可靠、高效、可伸缩的方式进行处理的。Hadoop 的核心框架为 HDFS（Hadoop Distributed File System）、MapReduce 和 HBase，最底部是 HDFS，HDFS 的上一层是 MapReduce 引擎，如图 2.2 所示。其中 HDFS 实现对分布式存储的底层支持，用于存储 Hadoop 集群中所有存储节点上的文件，HBase 则为大量非结构化数据存储和索引提供了条件，MapReduce 则实现对分布式并行任务处理的程序支持，能够让用户编写的 Hadoop 并行应用程序运行更加简化。

Hadoop 作为开源的云计算平台已经在互联网领域得到了广泛的应用，互联网公司往往需要存储海量的数据并对其进行处理，而这正是 Hadoop 的强项。如 Facebook 使用 Hadoop 存储内部的日志拷贝以及数据挖掘和日志统计；Yahoo 利用 Hadoop 支持广告系统并处理网页搜索；Twitter 则使用 Hadoop 存储微博数据、日志文件和其他中间数据等。在国内，Hadoop 同样也得到了许多公司的青睐，如百度主要将 Hadoop 应用于日志分析和

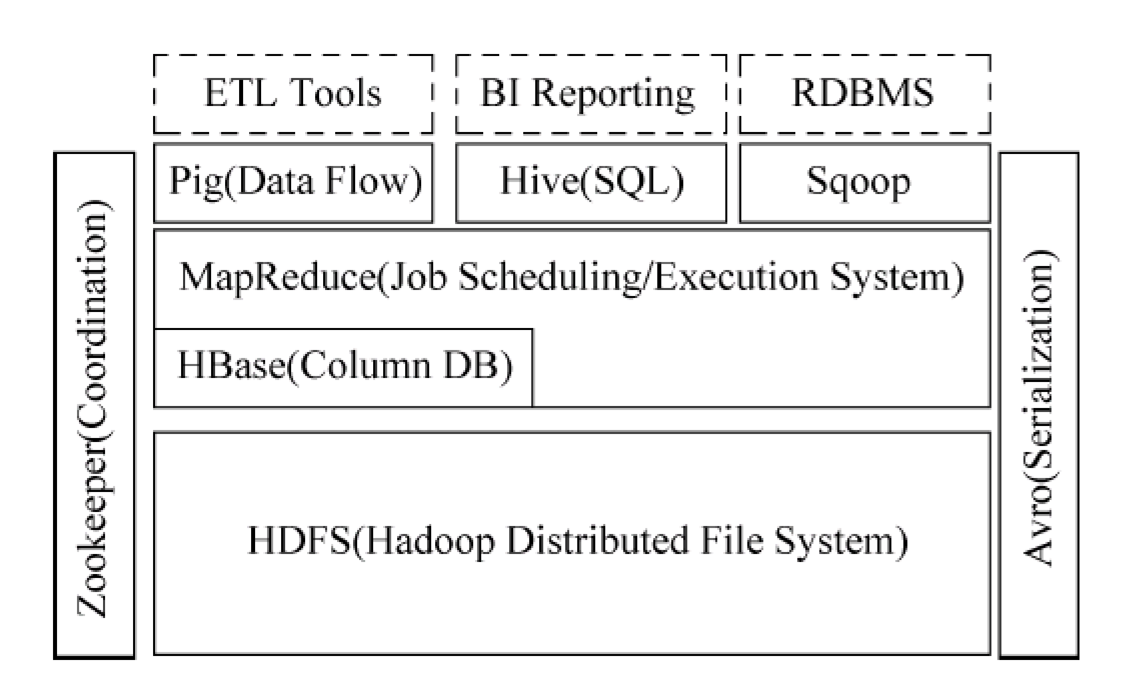


图 2.2　Hadoop 生态系统图

网页数据库的数据挖掘；阿里巴巴则将 Hadoop 用于商业数据的排序和搜索引擎的优化等。随着互联网的发展，新的业务模式还将不断涌现，Hadoop 的应用也会从互联网领域向电信、电子商务、银行、生物制药等领域拓展。

众所周知，现代社会的信息量增长速度极快，这些信息里又积累着大量的数据，其中包括个人数据和工业数据。预计到 2020 年，每年产生的数字信息将会有超过 1/3 的内容驻留在云平台中或借助云平台来处理。我们需要对这些数据进行分析和处理，以获取更多有价值的信息。那么如何高效地存储和管理这些数据？如何分析这些数据呢？这时可以选用 Hadoop 系统，它在处理这类问题时采用了分布式存储方式，提高了读写速度，并扩大了存储容量。采用 MapReduce 来整合分布式文件系统上的数据，可保证分析和处理数据的高效。与此同时，Hadoop 还采用存储冗余数据的方式保证了数据的安全性。

2.3.2　HDFS

HDFS(Hadoop Distributed File System)是 Hadoop 的一个分布式文件系统，由于其具有高容错性(Fault-Tolerant)的特点，因而可以设计部署在低廉(Low-Cost)的硬件上，并能以高吞吐率(High Throughput)来访问应用程序的数据，适合那些有着超大数据集的访问。

一个 HDFS 文件系统由一个主控节点(NameNode)和一组从节点(DataNode)构成，主控节点为一个管理整个文件系统的主服务器，包括文件系统命名空间及元数据的管理、对文件访问请求的处理。从节点是对数据块进行实际的存储和管理。主控节点向从节点分配数据块，建立数据块和从节点的对应关系；从节点处理系统用户对数据的读写请求以及主控节点对数据块的创建、删除副本的指令。一个集群设置唯一的 NameNode，这样在很大程度上简化了系统的架构。

HDFS 以分布式的存储方式存储大数据，可扩展性较好，容错能力、数据吞吐能力及并发访问能力都相当强大，为上层大数据的处理应用程序提供了强大的数据存储和访问功能支撑。而且 HDFS 放宽了可移植操作系统接口(Portable Operating System Interface, POSIX)的要求，实现了以流的形式访问文件系统中的数据。HDFS 原本是开源的 Apache 项目 Nutch 的基础结构，最后它成为了 Hadoop 的基础架构之一。

2.3.3　HBase

HBase 位于结构化存储层，是一个分布式的 NoSQL 数据库，它建立在 HDFS 之上，为大规模的结构化、半结构化及非结构化的数据提供了实时读写及随机访问功能。对于海量数据的存储，HBase 对硬件的要求不高，普通的服务器集群就能做到。HBase 对数据进行查询增改等操作的性能比较高，一般情况下都与数据量的大小无关，即使表中的数据记录非常大，对某条记录的查询也可以快速地完成。HBase 对数据模型的定义非常灵活，它采用的是列式存储而不是基于行的模式，表为一个分布式多维表，包括行关键字（Row Key）、列族（Column Family）、列名（Column Name）和时间戳（Timestamp）。字段数据对应的键值对为：

{row key，column family，column name，timestamp}→value

根据 row key、column key 和 time stamp 对数据进行查询，时间戳使得同一份数据有多个版本。一般情况下，查询数据的条件是基于列名的。与传统行存储方式的数据库不同的是，HBase 不需要扫描所有行的数据，这在很大程度上提高了数据访问性能。此外，HBase 还具有对数据读写严格一致性（即保证读到的是最新的数据）、高效的随机读写能力、较好的可扩展性等优点。

2.3.4　Hive

Hive 是一个基于 Hadoop 的数据仓库，它提供了一种类似 SQL 查询语言的编程接口，可以运用 HiveQL 语言对数据进行查询分析等操作，避免了复杂的 MapReduce 程序的设计编写，在很大程度上降低了对数据进行查询分析时应用程序的开发。Hive 是基于 HDFS、HBase 和 MapReduce 工作的，Hadoop 大数据平台从 Hive 接收数据处理指令，通过 HDFS、HBase，并配合 MapReduce 完成操作。Hive 不仅能够使用分区（Partition）及桶（Bucket）对数据进行存储以提高数据查询的性能，而且在写入数据时不检查数据的类型，从而达到高速加载数据的目的，这种模式能够满足大规模数据的需求。Hive 能结合 ETL 工具导入导出数据，在传统数据库和 Hadoop 平台之间充当桥梁的作用，使两者之间的联系更为紧密。

2.3.5　MapReduce

MapReduce 是由 Google 提出的，是一种面向大数据并行处理的计算模型，对计算数据和计算任务能够自动完成并行化处理。MapReduce 提供了简单方便的并行程序设计方法，用函数 Map 和函数 Reduce 编程实现并行计算，编程人员在不了解分布式并行编程的情况下也能方便地将自己的程序运行在分布式系统上，使得大数据的编程和计算变得更加简便。MapReduce 采用的是“分而治之”的思想，将大规模数据处理任务划分成若干个子任务进行处理，再将结果进行合并得到计算结果，从而完成大数据的并行化处理。MapReduce 定义了 Map 和 Reduce 两个编程接口：

```
map:(k1;v1)→[(k2;v2)]
```

```
reduce:(k2;[v2])→[(k3;v3)]
```

其中参数都是键值对形式的数据。Map 和 Reduce 处理的过程如下：数据以键值对的形式(k1;v1)传入 map 函数，经 map 函数处理后生成中间键值对[(k2;v2)]，然后对这些中间键值对进行处理，得到键值对(k2;[v2])，其中[v2]代表相同键 k2 的不同值 v2 的集合，将其传入 reduce 函数，经 reduce 函数处理后最终以键值对[(k3;v3)]的形式输出。Map 和 Reduce 两个阶段都是并行处理的。

2.3.6 Strom

Twitter Storm 是一个免费、开源的分布式实时计算系统，它可以简单、高效、可靠地处理大量的流数据。在 Twitter 中进行实时计算的系统就是 Storm，它在数据流上进行持续计算，并且对这种流式数据处理提供了有力保障。

Storm 运行于集群之上，与 Hadoop 集群类似。但在 Hadoop 上运行的是 MapReduce Jobs，而在 Storm 上运行的是 Topologies。两者大不相同，一个关键区别是 MapReduce 的 Job 最终会结束，而 Topology 永远处理消息(或直到 kill 它)。Storm 将数据以 Stream 的方式，并按照 Topology 的顺序依次处理并最终生成结果。

Storm 对一些概念进行了抽象化，其主要术语和概念包括 Streams、Spouts、Bolts、Topology 和 Stream Groupings。Topology 生态系统图如图 2.3 所示，数据从源头 Spout 中进入，依照拓扑顺序，使用相应的 Bolt 依次处理，得到最终数据。这种由各个用户自定义的处理器(Bolt)组合而成的拓扑结构能够适应大多数的业务需求。因此，只要业务能够组合成相应的拓扑逻辑，就能够借助 Storm 框架，也就无需考虑实时计算的低延迟、高性能、分布式、可扩展、容错等问题了。

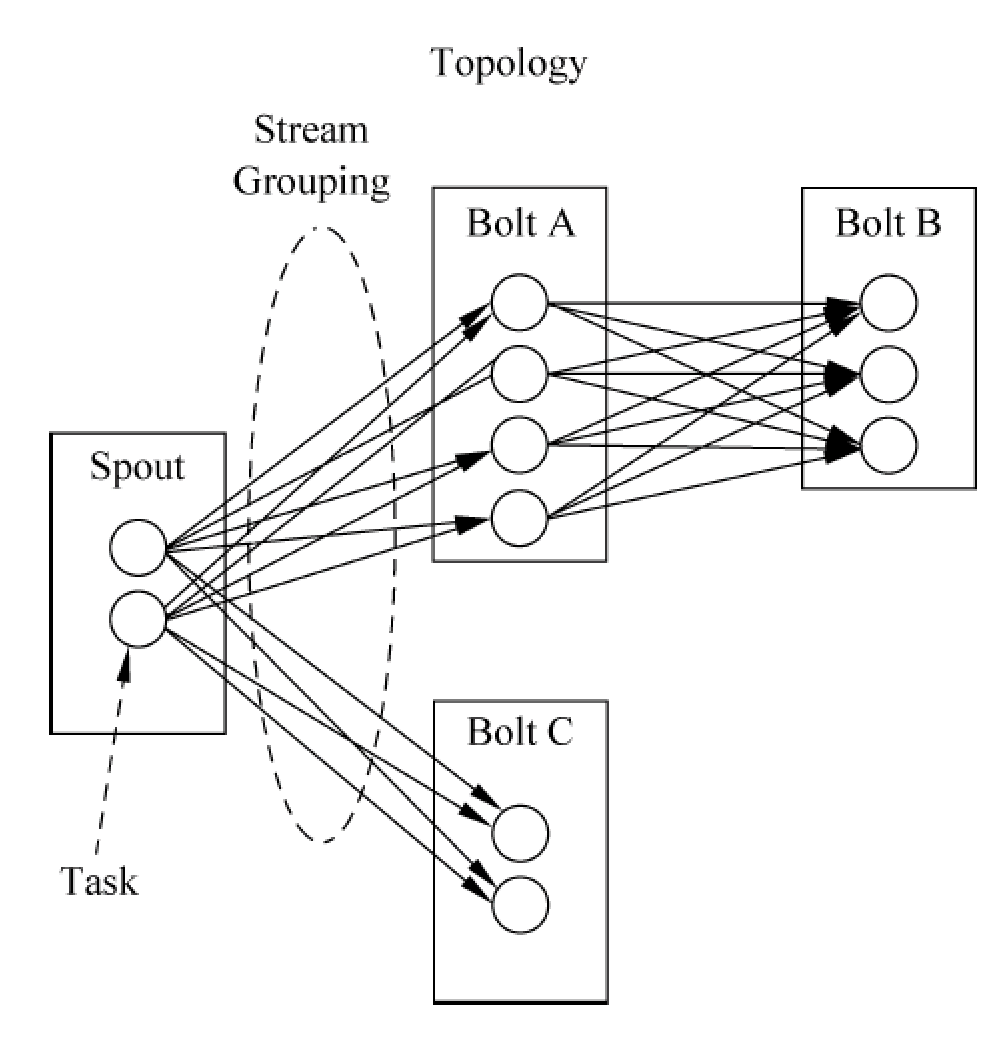


图 2.3 Topology 生态系统图

流 Stream 是一个不间断的无界的连续 Tuple(元组，是元素有序列表)，这些无界的元组会以分布式的方式并行地创建和处理。每个流 Stream 都有一个源 Spouts，Spouts 会从

外部读取流数据并发出 Tuple。流的中间状态抽象为 Bolts，Bolts 可以处理 tuples，同时它也可以发送新的流给其他 Bolts 使用。Bolts 作为消息处理者，处理输入的数据流并产生输出的新数据流。Bolts 中可执行过滤、聚合、查询数据库等操作。

为了提高效率，在 Spout 源可以接上多个 Bolts 处理器。Storm 将这样的无向环图抽象为 Topology，Topology 是 Storm 中最高层次的抽象概念。当 Spout 或者 Bolt 发送元组到流时，它就发送元组到每个订阅了该流的 Bolt 上进行处理。

对比 Hadoop 的批处理，Storm 是一个实时的、分布式及具备高容错的计算系统。同 Hadoop 一样，Storm 也可以处理大批量的数据。然而 Storm 在保证高可靠性的前提下还可以让处理进行的更加实时，通常被比作“实时的 Hadoop”。也就是说，所有的信息都会被处理。Storm 同样还具备容错和分布计算这些特性，这就让 Storm 可以扩展到不同的机器上进行大批量的数据处理。

Storm 流处理技术作为大数据处理技术之一，其应用场景也有很多，总的来说，一方面可应用于处理金融服务如股票交易、银行交易等产生的大量实时数据；另一方面主要应用于各种实时 Web 服务中，如搜索引擎、购物网站的实时广告推荐，SNS 社交类网站的实时个性化内容推荐，大型网站、网店的实时用户访问情况分析等。实时的数据计算和分析对于大型网站来说具有重要的实际意义，不仅可用于网站的实时业务监控，也可以实现用户实时个性化内容推荐等。

随着企业数据量的迅速增长，存储和处理大规模数据已成为企业的迫切需求。在大数据的计算中，开源分布式离线处理技术 hadoop 与提供了分布式流处理框架的 storm 在各大领域都得到了广泛的应用。

2.4　大数据分析与 R

随着大数据时代的到来，大量来自互联网、金融、生物等领域的数据需要分析处理。Hadoop 的分布式数据处理模式让原来不可能的 TB、PB 级数据量计算成为了可能，而 R 语言的强大之处在于其统计分析功能，但是对于大数据的处理，R 受到内存和性能方面的限制，只能通过抽样进行计算分析。为了克服这一不足，使得 R 能够有效地处理大数据，RHadoop 应运而生。RHadoop 是由 Revolution Analytics 公司发起的一个开源项目，它使得 Hadoop 集群上存储在 Hadoop 分布式文件系统中的数据可以实现本地 R 分析，并对这些计算结果进行整合，这一举措使得大规模数据得到了有效的管理，同时充分利用了 R 强大的数据分析功能。由此可以看出，这两种技术的结合既是产业界的必然导向，也是产业界和学术界的交集，更为交叉学科的人才提供了无限广阔的想象空间。

2.4.1　RHadoop

RHadoop 包含三个 R 包：rhdfs、rmr 及 rhbase，它们分别对应 Hadoop 架构中的 HDFS、MapReduce 和 HBase 三个部分。RHadoop 的三个包为 R 进行大数据的分析操作提供了以下功能：

rhdfs 包的主要功能是调用 HDFS API 对存储在 HDFS 上的数据进行操作，使得 R 对分布式数据文件的操作变得更加简便。

rmr 包为 R 提供了 Hadoop MapReduce 功能，在 R 上需要将程序分成两个阶段(Map 阶段和 Reduce 阶段)，然后调用 rmr 实现任务的提交，进而调用 Hadoop streaming 的 MapReduce API，从而在集群上分布式地实现 R MapReduce 程序，完成 R 对大数据的分析操作。

rhbase 包为 R 提供了 HBase 数据库管理功能，在 R 环境中实现对 HBase 中数据的读写查询等操作。

1. Rhdfs 包的基本操作函数

- hdfs. init：初始化 rhdfs，该操作的语法为 hdfs. init()。
- hdfs. defaults：获得 rhdfs 的默认设置，该操作的语法为 hdfs. defaults ()。
- hdfs. put：从本地文件系统复制文件到 HDFS 系统中。例如将存放于本地文件系统中的 sample. txt 文件复制到 HDFS 中，操作如下：

hdfs. put('/local/hadoop/sample. txt','/RHadoop/first')

- hdfs. copy：从 HDFS 目录复制文件到本地文件系统，例如：

hdfs. copy('/RHadoop/first','/RHadoop/second')

- hdfs. move：移动文件，将文件从 HDFS 文件夹移动到另一个 HDFS 文件夹中，例如：

- hdfs. move('/local/hadoop/sample. txt','/RHadoop/first')
- hdfs. rename：重命名文件，在 R 环境中重命名存储在 HDFS 中的文件，例如：

hdfs. rename('/RHadoop/first/ sample1. txt',' /RHadoop/first/ sample2. txt')

- hdfs. delete/ hdfs. rm/hdfs. rmr：在 R 中将 HDFS 中的文件或文件夹删除，例如：

hdfs. delete('/RHadoop')

- hdfs. chmod：修改文件权限，例如：

hdfs. chmod('/RHadoop',permissions="777")

- hdfs. file：初始化文件，使得在 HDFS 中的文件能够进行读写操作，例如：

F=hdfs. file('/Rhadoop/first/sample. txt',mode="r",buffersize=5242880,overwrite=TRUE)

- hdfs. write：写入文件，将 R 对象通过 streaming 写入存储在 HDFS 中的文件，语法为 hdfs. write(object,con,hsync)。

其中，object 为写入磁盘的 R 对象，con 为已初始化能够进行读写操作的文件。如果参数 hsync 设置为 TRUE，那么写入对象后该文件将被同步。

- hdfs. close：关闭读写流，关闭后不能对文件进行读写。例如，关闭初始化的 HDFS 文件 F，hdfs. close(F)。
- hdfs. read：从 HDFS 中的文件读取文件内容，语法为：hdfs. read(con,n,start)。

其中，con 为已初始化能够进行读写操作的文件，n 为所读取的字节数，start 为开始读取的位置，默认为当前位置。

• hdfs. dircreate/hdfs. mkdir：创建文件夹，用于在 HDFS 中创建文件夹，例如：

hdfs. dircreate("/Rhadoop/1/")

• hdfs. ls：将 HDFS 中的文件夹内容列出来，例如：

hdfs. ls("/tmp")

• hdfs. file. info：获取 HDFS 文件的元信息，例如：

hdfs. file. info("/tmp")

2. rmr 包的基本操作函数

• to. dfs：向 HDFS 文件系统中写入 R 对象。

• from. dfs：从 HDFS 文件系统中读取 R 对象。

• mapreduce：定义、执行 MapReduce 任务，函数语法如下：

mapreduce(input, output, map, reduce, combine, input. format, output. format, verbose)

• keyval：创建、提取键值对，函数语法为 keyval(key, val)。

2.4.2 RHIPE

RHIPE(R and Hadoop Integrated Programming Environment，Hadoop 和 R 集成编程环境)是一种 R 和 Hadoop 的结合技术，使用 Divide 和 Recombine 技术实现大数据分析。RHIPE 在 R 和 Hadoop 之间充当桥梁的作用，它使得 R 分析大规模数据得以实现，可以在 R 上操作 MapReduce 程序。

2.4.3 RHive

RHive 是一款通过 R 语言直接访问 Hive 的工具包，是由 NexR 公司研发的。通过使用 RHive 可以在 R 环境中写 HQL(HiveQL)，将 R 的对象传入到 hive 中，在 hive 中进行计算。在 RHive 中小数据集在 R 中执行，大数据集在 hive 中运行。

RHive 的一些基本操作如下：

• rhive. init：初始化 Rhive，语法为 rhive. init()。

• rhive. connect：连接 hive 服务器，例如：

```
rhive.connect("192.168.85.105")
```

• rhive. list. tables：列出所有的表，语法为 rhive. list. tables()。

• rhive. desc. table：查看某个表的结构，例如：

```
rhive.desc.table('hivetable')
```

• rhive. query：执行 HQL 查询操作，例如：

```
rhive.query("select * from hivetable")
```

• rhive. close：断开与 hive 服务器的连接，语法为 rhive. close()。

2.4.4 RHBase

RHBase 依赖于 Hadoop、HBase 和 Thrift，通过 RHBase 可以实现从 HBase 将数据加载到 R 中，包括新建或删除表、显示表结构、读取或插入数据等操作。

RHBase 的一些基本操作：

- hb.list.tables：列出所有 HBase 表，语法为 hb.list.tables()。
- hb.new.table：创建 HBase 表，例如：

```
hb.new.table("hbasetable")
```

- hb.describe.table：显示表结构，例如：

```
hb. describe.table("hbasetable")
```

- hb.get：读取表中的数据，例如读取 hbasetable 表中行关键字为 1001 的字段：

```
hb.get("hbasetable","1001")
```

- hb.insert：向表中插入数据，语法为：

```
hb.insert(tablename,changes)
```

其中，tablename 为需要插入数据的 HBase 表，changes 为插入的字段内容，为列表的形式。

- hb.delete.table：删除 HBase 表，例如删除表 hbasetable：

```
hb.delete.table("hbasetable")
```

- hb.delete：删除表中的字段，例如删除表 hbasetable 中行关键字为 1001 的字段：

```
hb.delete("hbasetable","1001")
```

2.5 国泰安的大数据

2.5.1 大数据实验室建设

国泰安大数据实验室可根据高校的实际需求和专业及人才定位情况，根据学科研究领域及方向、师资实验室设置、运行等基本情况，针对不同院校的特色、方向课程等工作的开展情况，设计大数据研究中心、实训基地来满足学校的具体需求。

从顶层思路的大数据价值链出发，借鉴大数据通用架构图，结合市场常用软件及国泰安自有的软件形成了国泰安大数据实验室解决方案。为大数据实验室的建设提供数据源、大数据采集与 ETL、大数据存储、大数据分析与挖掘、大数据展示与可视化 5 大模块全面系统的服务。图 2.4 是国泰安大数据实验室解决方案。

表 2.3 为国泰安大数据实验室的软件配置列表，其中包含了从数据源、大数据采集、ETL、存储、分析挖掘到可视化展示整个系统的软件配置。

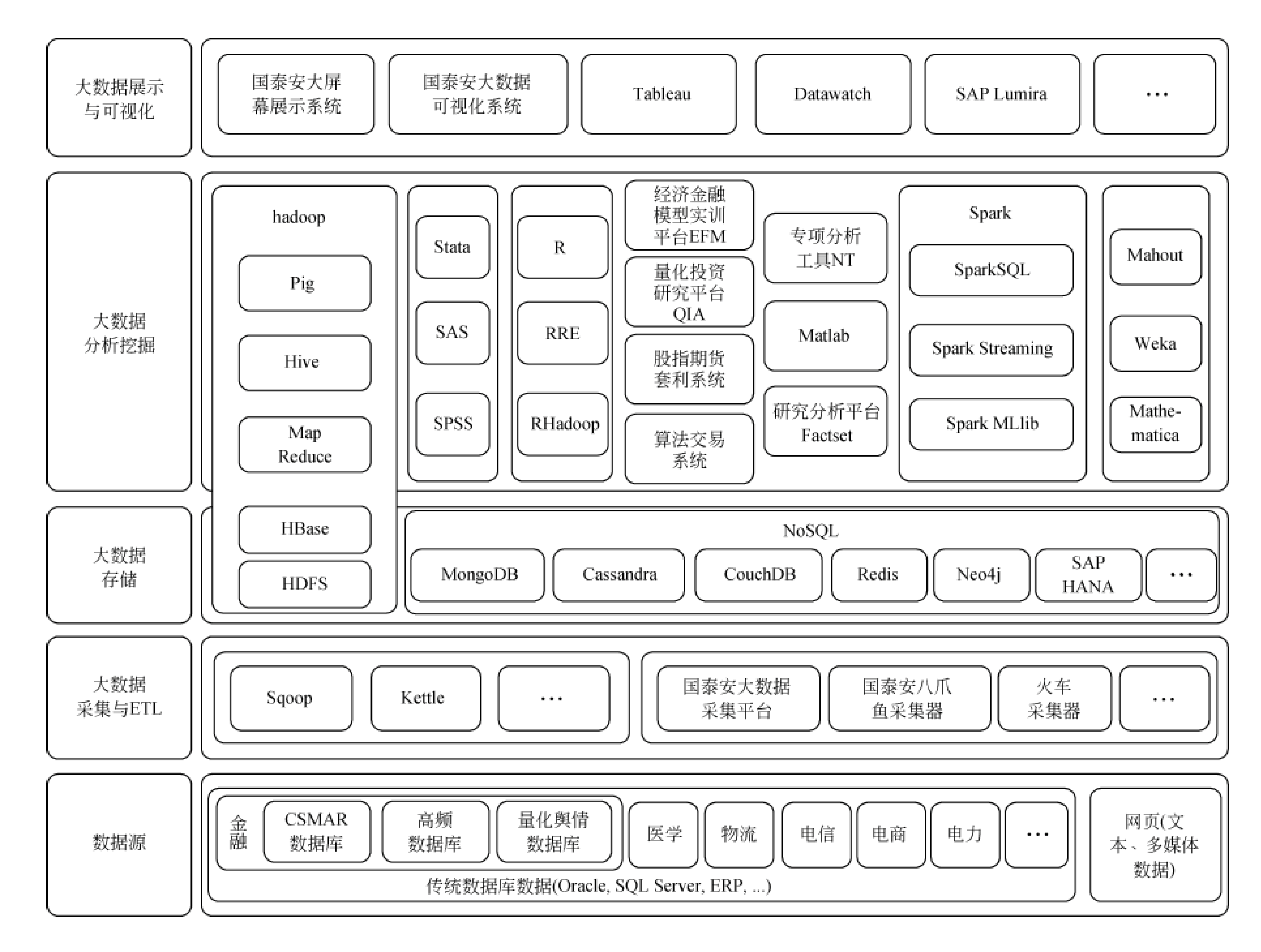


图 2.4　国泰安大数据实验室解决方案

表 2.3　大数据实验室软件配置列表

分　　类	软件名称	简　　介
数据源	CSMAR 数据库	CSMAR 数据库是专门针对中国金融、经济领域的研究型精准数据库，包括股票市场、公司研究、基金市场、债券市场、衍生市场、经济研究、行业研究、海外研究和专题研究等 11 个大系列，75 个数据库
	量化舆情数据库	量化舆情数据库是为了支持新闻传媒、品牌管理和量化投资等研究，通过接收新闻站点、论坛、博客和微博等海量舆情数据而建设的数据存储系统
	高频数据库	高频数据库是包含股票、基金、债券、权证、股指期货、商品期货，港交所证券在内的各类高频数据，以及基于高频数据传输、更新、应用软件在内的一套整体的系统解决方案
大数据采集与 ETL	国泰安大数据采集平台	国泰安大数据采集平台实现对各类不同的数据源的手工、半手工、结构化、非结构化和半结构化数据进行统一采集管理
	国泰安八爪鱼采集器	国泰安八爪鱼数据采集系统以完全自主研发的分布式云计算平台为核心，可以在很短的时间内，轻松地从各种不同的网站或者网页获取大量的规范化数据
	火车采集器	火车采集器是一款专业的网络数据采集/信息挖掘处理软件
	Sqoop	Sqoop 是一款开源的工具，主要用于在 Hadoop(Hive)与传统的数据库(mysql、postgresql,…)间进行数据的传递。Sqoop 是用来将 Hadoop 和关系型数据库中的数据相互转移的工具
	Kettle	Kettle 是一款国外开源的 ETL 工具，纯 Java 编写，可以在 Windows、Linux 和 UNIX 上运行，数据抽取高效稳定

续表

分类	软件名称	简介
大数据存储	HBase	HBase是一个分布式的、面向列、适合于非结构化数据存储的开源数据库，是一个数据库
	HDFS	HDFS是一个分布式文件系统，是Hadoop体系中数据存储管理的基础
	MongoDB	MongoDB是一个高性能、开源、无模式的文档型数据库，是当前NoSQL数据库中比较热门的一种
	Cassandra	Cassandra是一套开源分布式NoSQL数据库系统，是一种流行的分布式结构化数据存储方案
	CouchDB	CouchDB是一个开源的面向文档的数据库管理系统，可以通过RESTful JavaScript Object Notation (JSON) API访问
	Redis	Redis是一个开源的使用ANSIC语言编写、支持网络、可基于内存也可持久化的日志型、Key-Value数据库
	Neo4j	Neo4j是一个嵌入式、基于磁盘的、支持完整事务的Java持久化引擎，它在图(网络)中而不是表中存储数据
	SAP HANA	SAP HANA是一款完备的实时分析解决方案
大数据分析挖掘	MapReduce	MapReduce是一种计算和编程模型，用于大规模数据集(大于1TB)的并行运算
	Hive	Hive是由Facebook开发的建立在Hadoop上的数据仓库基础构架，是用来管理结构化数据的中间件
	Pig	Pig是一个基于Hadoop的大规模数据分析平台，包含Pig Interface和Pig Latin两个部分，其中Pig Latin语言的编译器会把类SQL的数据分析请求转换为一系列经过优化处理的MapReduce运算
	R	R，一种自由软件编程语言与操作环境，具有统计分析功能
	RHadoop	RHadoop是一款Hadoop和R语言结合的产品，由Revolution Analytics公司开发，可对海量数据进行分析
	RRE	Revolution R Enterprise拥有各种各样的数据可视化、统计分析、预测性建模及机器学习的能力，能得到最具成本效率的分析，并能快速地分析大数据，完全与R语言兼容
	经济金融模型实训平台(EFM)	经济金融模型实训平台是一个集经济金融数理统计模型教学、建模、实训、交流、应用为一体的开放式教学实训平台
	Matlab	Matlab是matrix&laboratory两个词的组合，意为矩阵工厂(矩阵实验室)。Matlab是由美国mathworks公司发布的主要面对科学计算、可视化及交互式程序设计的高科技计算环境
	Spark	Spark是UC Berkeley AMP lab所开源的类Hadoop MapReduce的通用的并行计算框架，基于map reduce算法实现分布式计算
	Spark Streaming	Spark Streaming是建立在Spark上的实时计算框架，通过它提供的API和基于内存的高速执行引擎，用户可以结合流式、批处理和交互式查询应用
	Spark MLlib	MLlib是Spark对常用的机器学习算法的实现库，同时包括相关的测试和数据生成器
	SparkSQL	Spark SQL是支持在Spark中使用Sql、HiveSql、Scaca中的关系型查询表达式

续表

分类	软件名称	简介
大数据分析挖掘	Mahout	一个用于机器学习和数据挖掘的分布式框架，区别于其他的开源数据挖掘软件，它是基于 Hadoop 之上的
	Weka	Weka 即怀卡托智能分析环境，是基于 Java 环境下开源的机器学习及数据挖掘软件
大数据展示与可视化	国泰安大数据可视化系统	BI 可视化开发工具，支持各种数据源、丰富的图表类型
	国泰安大屏幕管理系统	一个采用独特的开发理念，突破了传统软件的各种瓶颈、技术创新的应用型高端软件系统
	Datawatch	一款用于实时数据处理、分析和数据可视化的软件
	Tableau	Tableau 是桌面系统中最简单的商业智能工具软件

2.5.2 大数据分析平台

国泰安金融大数据实验室由金融大数据采集、金融大数据内容与存储、金融大数据分析与挖掘、金融大数据展示与可视化、金融大数据智慧教学平台 5 大模块组成。其中，大数据分析平台(big data analysis，BDA)是一个面向大数据分析的教学系统。BDA 是集经济金融、数理统计、数据分析模型的教学、建模、实训、交流、应用为一体，基于 R 的开放式教学实训平台，是一个通过提供专业的大数据分析领域常用的算法模型，用统计分析领域中应用范围最广的 R 语言来实现，为学校师生提供算法模型的教与学的软件平台。DBA 由以下模块组成。

(1) 模型演示模块：对模型从理论到实战程序、数据结果、图形的全面展示。

(2) 模型 DIY 模块：自主创建模型的背景知识及模型的程序。

(3) 模型管理模块：实现对模型的修改、导出及同步功能。

(4) 教学管理模型：实现个人的在线作业，作业的发放和评阅等功能。

(5) 编程 ABC 模块：分享你我他的资源，实现在线学习交流。

BDA 的主要功能和特点如下。

1. 全方位式的教学

教学内容模型化、图形化、数字化，包括模型的理论介绍及详细讲解步骤，每个步骤的推导及演算说明、参数说明、数据、程序，以及每步的计算结果和图形。

2. 可视化、流程化的建模

标准化、模块化的建模流程和框架，可修改原有的案例程序及数据，也可自主建模；可使用自有的 Excel、txt、RData、csv、mat 格式数据计算建模，也可使用 API 调用 CSMAR 数据库的数据进行建模。对模型最多有 10 个步骤进行灵活分解。

3. 可视化编程

从数据到程序，从中间变量到目标结果，全流程可视化，轻松有趣地学习 R 编程。

4. 全面的教学管理平台

教师可在线完成按班级或指定的学生群体发放模型作业，查看作业模型程序、数据和结果，以及在线交流及辅导，随时随地实现作业的收发、评阅及辅导。

5. 引导式的学习平台

模型难易分明，既可满足基本的教学需求，又能够实现学生对软件、理论模型的编程开发学习，最重要的是平台上有非常贴近市场的真实案例开源代码资源学习，相信能够使学者完成从基础学习到职场准备的阶梯提升计划。

6. 金融建模大赛

每年两次的金融建模大赛，能够激发学生的创新和思考问题的能力。通过BDA建模，能够使读者更加规范化自己的程序，规范文章内容。

BDA的亮点主要体现在以下几个方面。

(1) 模型分步演示：使得学生学会更加清晰的逻辑编程思路。

(2) 丰富的理论背景知识：深刻了解模型背景、应用场景，拓宽学生解决问题的思路。

(3) R语言模型新建编译：开放编译接口、面向大数据分析领域的通用型语言。

(4) 图形结果展示：一个页面展示多个图形结果、数据结果，方便用户进行结果分析。

(5) 作业管理系统：提供教学的管理平台，将课程模型资源在统一的平台上进行管理。

PART

2

第二部分

R 语 言

CHAPTER 3

第3章 R语言简介

R 是一款开源的、专业的统计分析软件，是集数据分析、绘图、数据挖掘于一体的编程语言与操作环境。R 凭借强大的数据处理、数学统计分析等功能，以及免费自由的开源特性得到各类社会组织的青睐。本章详细介绍 R 的特性、基本功能及 R 包的获取使用等。

3.1　R 语言概述

R 语言是集数据分析与图形显示于一体的编程语言，是一种专业的统计分析软件。R 从根本上摒弃了套用模式的傻瓜式数据分析方式，它将数据分析的主动权和选择权交给使用者本身。数据分析人员可以根据问题的背景和数据的特点，更好地思考从数据出发如何选择和组合不同的方法，并将每一层输出反馈到对问题和数据处理的新思考上。R 为专业分析提供了分析的弹性、灵活性和扩展性，是利用数据回答问题的最佳平台。

R 语言主要有以下几个特点。

1. R 是自由软件

之所以称 R 是自由软件，是基于它的免费和开源。R 是一个用于统计计算的很成熟的免费软件，同时也能提供和其他同类型商业统计软件一样好的功能服务。R 还有一个亮点，即它是一款开源软件，用户可以和全球一流的统计专家合作讨论，也可以上传自己的软件包，可以说 R 是全世界统计学家思维的最大集中地。现如今，开放源代码的软件在科学研究和工程工作中越来越受到追捧。R 的开源性使得它从 20 世纪 90 年代被开发出来至今，一直在快速发展中。

2. R 的兼容性很好

R 的兼容性体现在两个方面：一方面，R 和其他程序设计语言的语法表述相似，使得有一定编程基础的人学习起来容易，并且它也是彻底地面向对象的统计编程语言，非常容易理解和使用；另一方面，R 可以实现与 Excel、SAS、SPSS 等常用统计软件的数据转换，也可以方便地插入由 C 语言等编制的计算机程序，这对数据整合工作非常有用。

3. R 是数据可视化的先驱

R 软件提供了非常丰富的 2D 和 3D 图形库，是数据可视化的先驱，能够生成从简单到复杂的各种图形，甚至可以生成动画，满足不同信息展示的需要。

4. 不断更新的加载包

Google 首席经济学家 Hal Varian 说："R 变得如此有用和如此快地广受欢迎是因为统计学家、工程师、科学家能够用它精炼代码或编写各种特殊任务的包。R 包增添了很多高级算法、作图颜色和文本注释，并通过数据库连接等方式提供了挖掘技术。金融服务部门对 R 表现出了极大的兴趣，各种各样的衍生品分析包相继出现。R 最优美的地方是它能够根据自己的需求修改很多前人编写的包的代码，实际上你是站在巨人的肩膀上。"

正是由于 R 具有免费、开源、模块多样齐全等众多特点，且在综合 R 档案网络(Comprehensive R Archive Network，CRAN)中提供了大量的第三方功能包，其内容涵盖了从统计计算到机器学习，从金融分析到生物信息，从社会网络分析到自然语言处理，从各种数据库、各种语言接口到高性能计算模型，可以说无所不包，无所不容，这也是为什么 R 获得越来越多各行各业的从业人员喜爱的一个重要原因。

类似 R 的统计软件种类有很多，最常见的有以下 5 种，它们有各自的优缺点。

(1) SAS：内容全面，价格昂贵，支持编程，是数据处理和统计分析的专用软件。

(2) SPSS：操作简单、无需编程、输出漂亮、功能齐全、价格合理，非统计专业人员的首选软件。

(3) Eviews：具有强大的多元回归和时间序列分析功能，计量专业首选软件。

(4) Matlab：功能强大的编程软件，矩阵运算快，统计分析功能较少，是数值计算和图像处理的首选软件。

(5) Excel：具有简单的统计分析功能，是商务办公软件。

这些软件的共同缺点：其一是"黑匣子"，即源代码不公开，只能运用已有功能，不能根据自身特殊需要进行修改；其二是"傻瓜软件"，对于一些简单分析，傻瓜式操作简便，适用于非统计专业人士，但是进行一些深入分析时就无法胜任或者步骤繁复。

3.2 R 的下载、安装和使用

3.2.1 RGui 界面

R 软件的获取及安装过程如下：

(1) 登录 R 语言官网(http://www.r-project.org/)，可以看到如图 3.1 所示界面。

(2) 单击左侧菜单栏的 CRAN 链接，进入图 3.2 所示页面，有一系列国家名称排序的镜像网站，选择与你所在地相近的网站。

(3) 依据自己计算机系统选择对应的下载，本书以 Windows 为例，如图 3.3 所示。

(4) 单击 base 链接进入图 3.4 所示页面，现在 R 更新到 3.1.1 版本，单击 Download R-3.1.1 for Windows 链接。

(5) 下载完成后双击程序文件进行安装，安装完成后便可以运行 R，界面如图 3.5 所示。

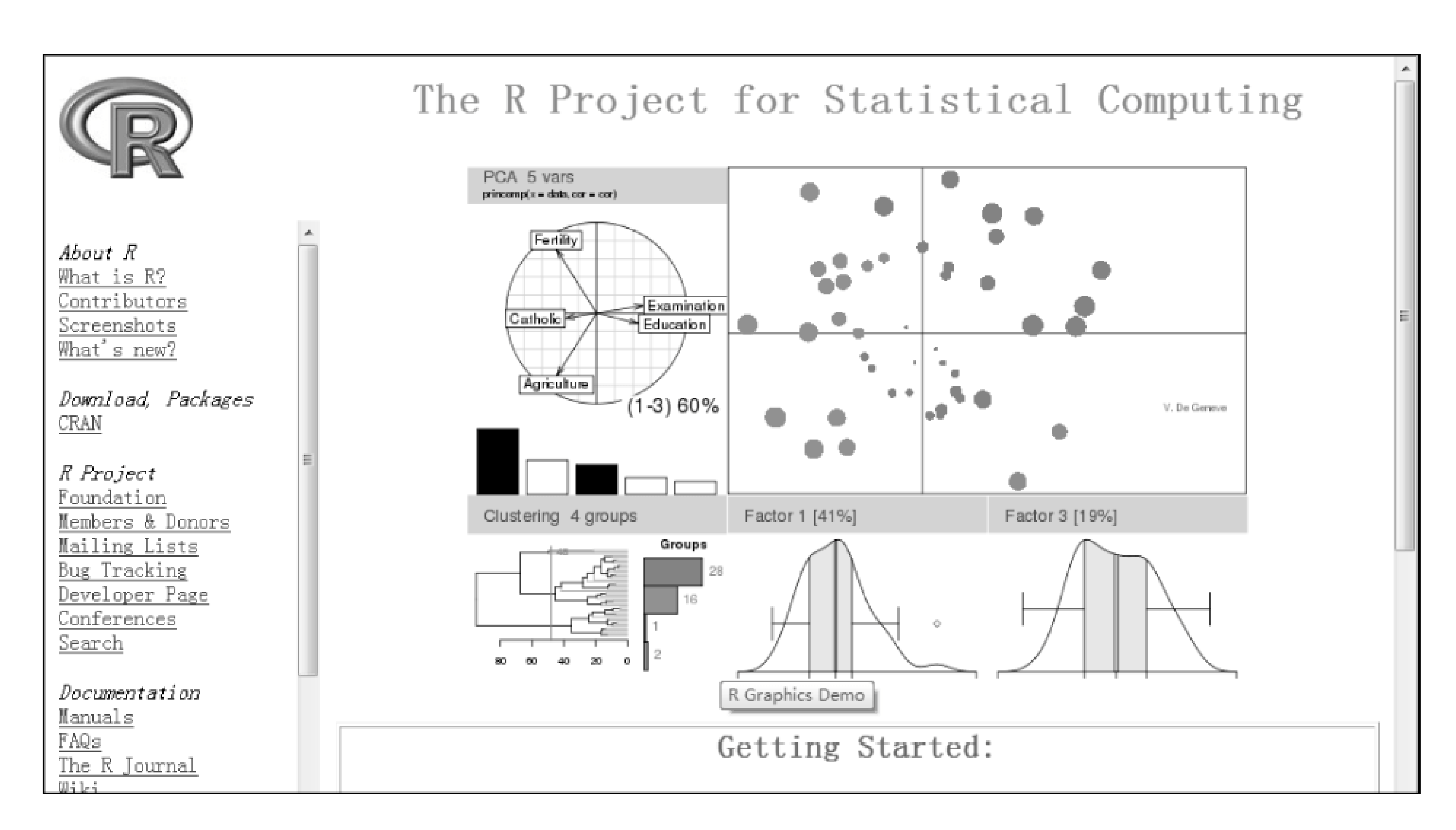


图 3.1　R 语言官网界面

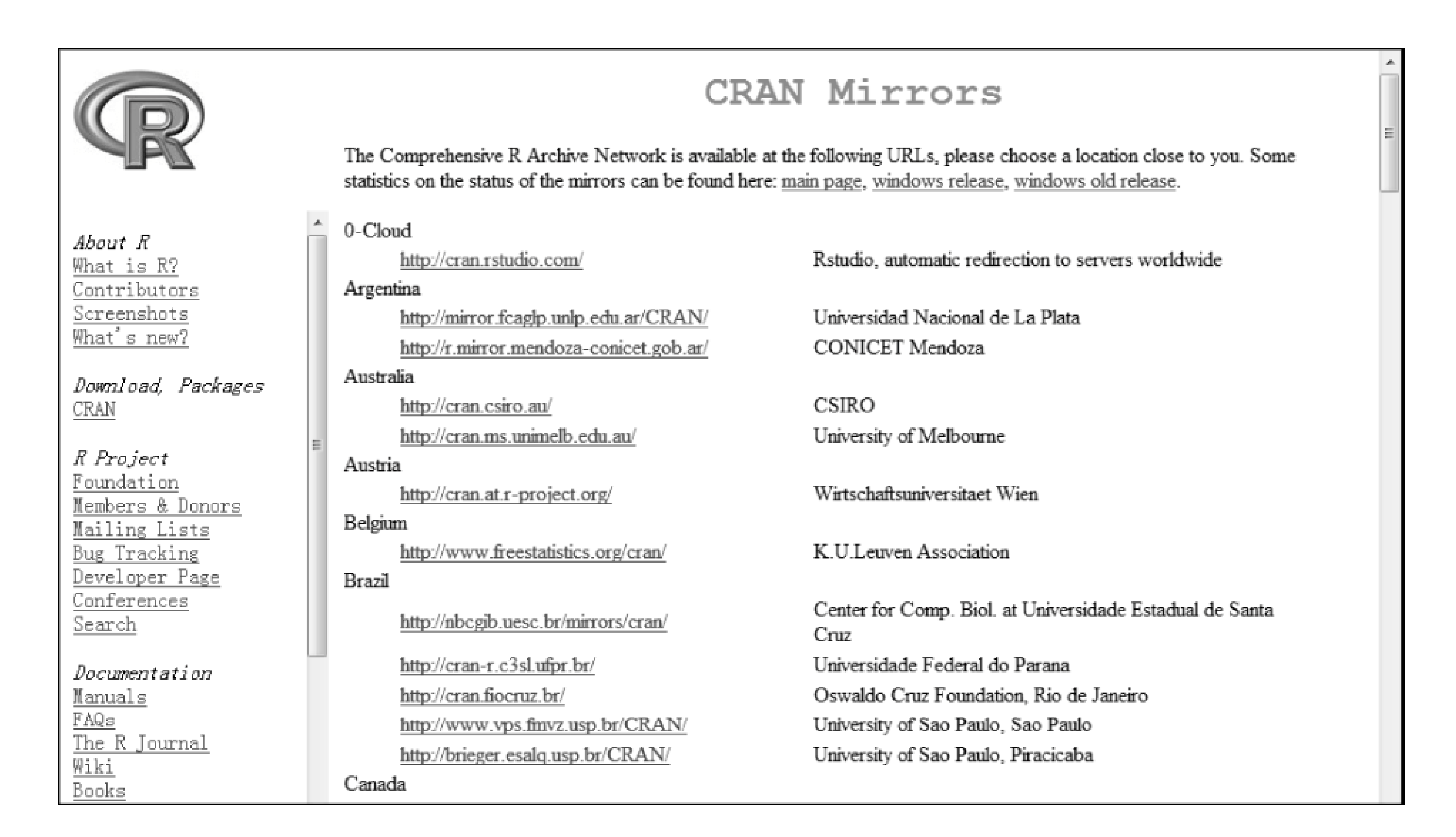


图 3.2　R 语言镜像网站选择界面

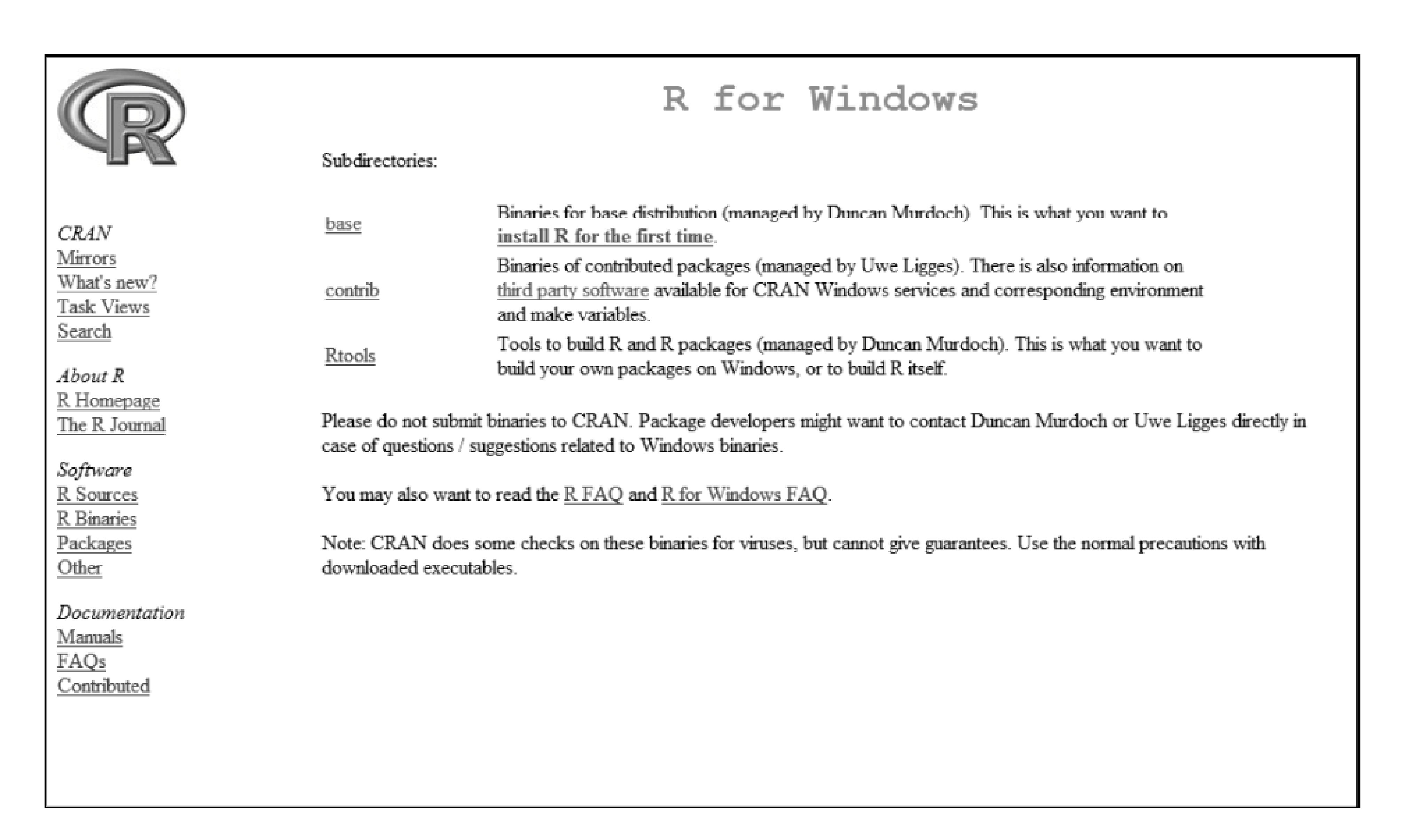


图 3.3　Windows 系统的 R 下载选择界面

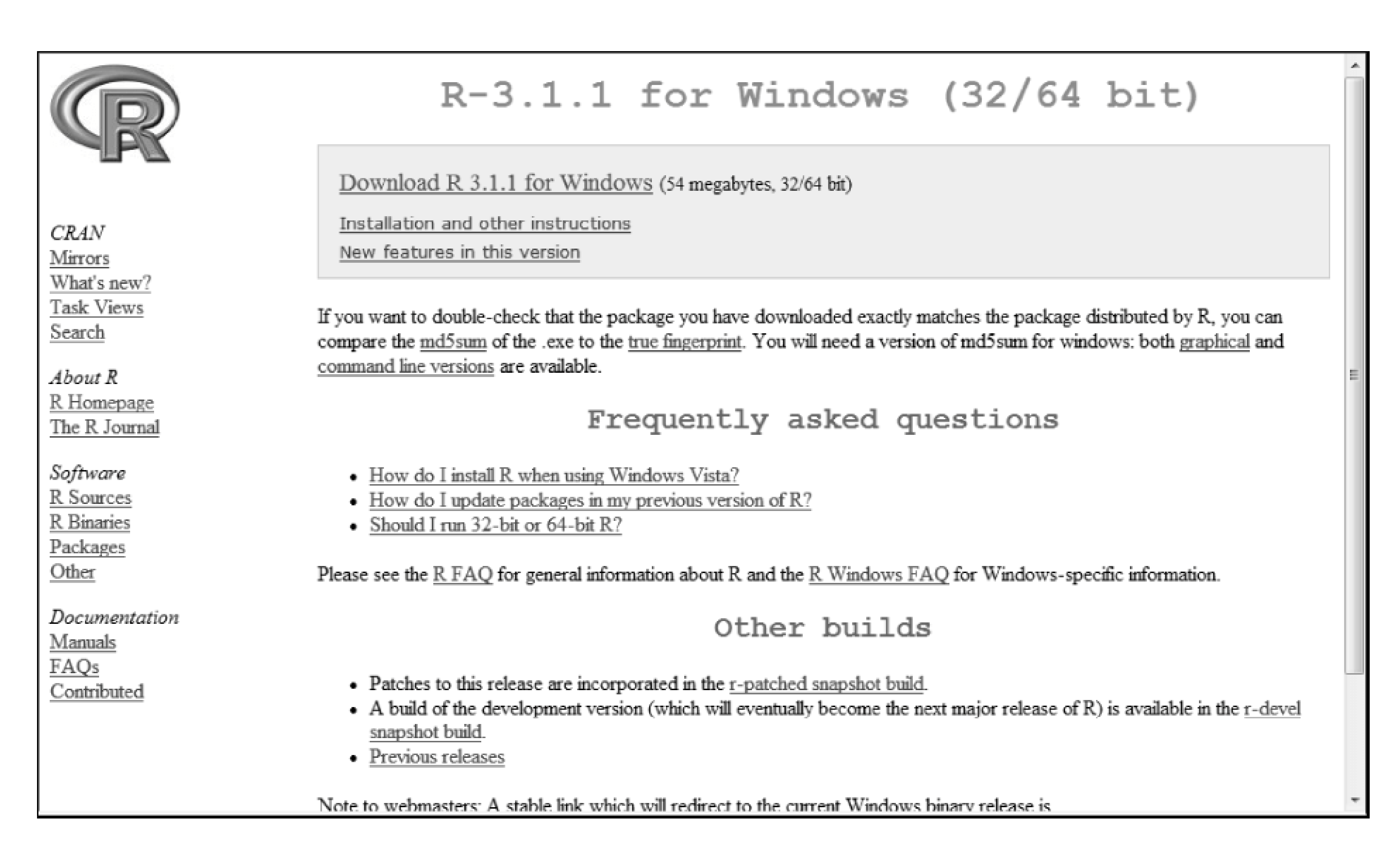


图 3.4　Windows 系统的 R-3.1.1 下载界面



图 3.5　R 运行界面

3.2.2　RStudio 界面

可以选择下载 RStudio，界面更加友好，设计更加人性化，建议读者下载使用，如图 3.6 所示。下载地址为 http://www.rstudio.com/。

运行 RStudio，可以看到它是由顶端的工具栏和 4 个小窗口组成，分别是文档编辑窗口、数据变量窗口、操作台窗口和结果展示窗口。

1. 文档编辑窗口

关于创建 R 文档，选择 File→New File→R Script 命令，或者按 Ctrl+Shift+N 组合键，就可以创建一个新的 R 文档，如图 3.7 所示。建议在使用 R 时创建一个文档进行编辑，可以保留自己的程序代码，以便出错时进行修改。

2. 数据变量窗口

给变量赋值以后会显示在此窗口，另外 RStudio 还提供了已安装软件包变量名和函数名查询。

3. 操作台窗口

在该窗口可以进行命令输入，数据结果也是在这里显示。

4. 结果展示窗口

各种酷炫的图表都将在此窗口展示，help 的内容也在这里显示，如果使用 R 语言，使用帮助(Help)命令将会弹出网页。RStudio 还提供了快速加载软件包的功能，在后面的章节会进行详细的说明。

5. RStudio 常用组合键

- Ctrl+L：清除控制台输出。

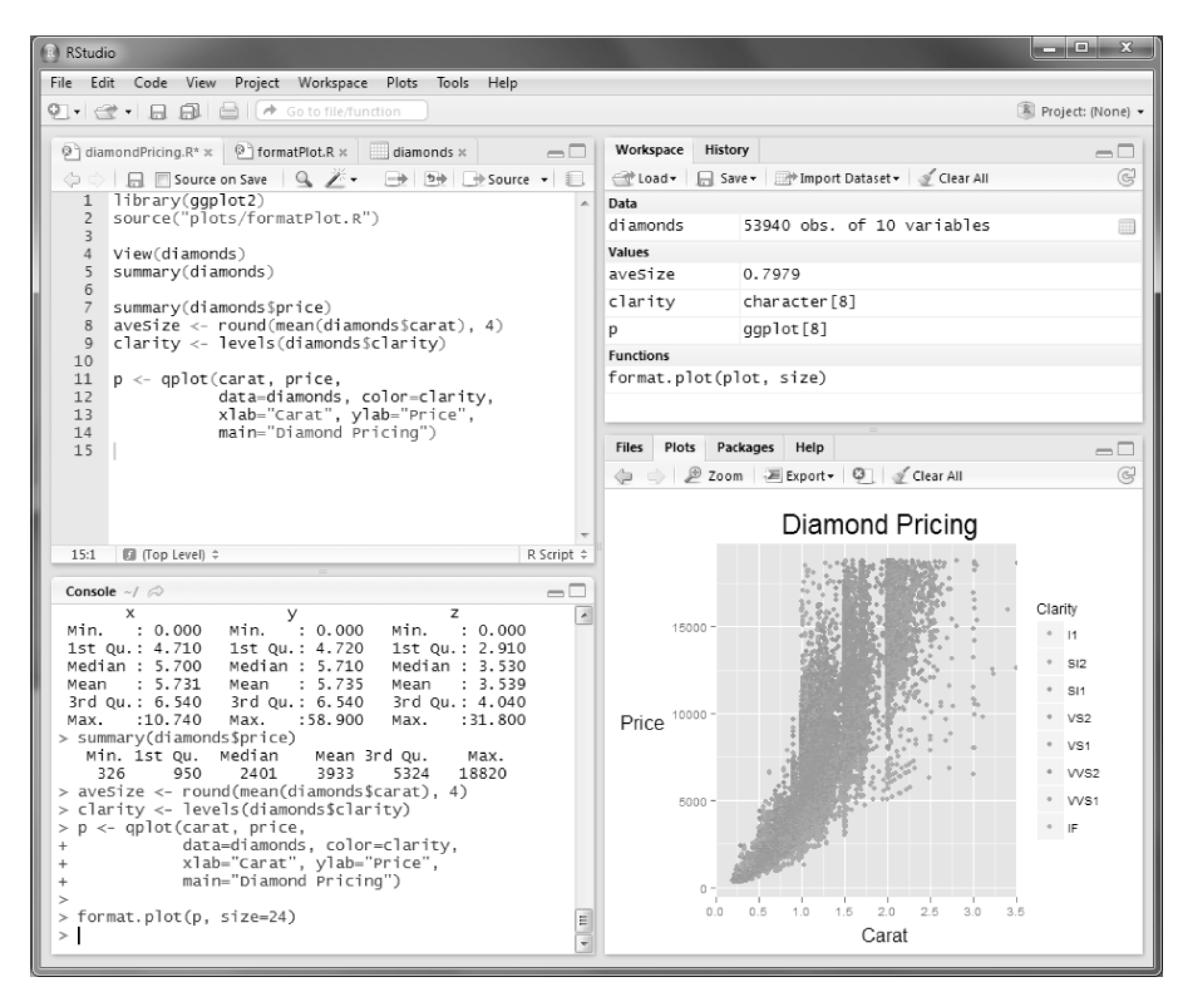


图 3.6 RStudio 操作界面

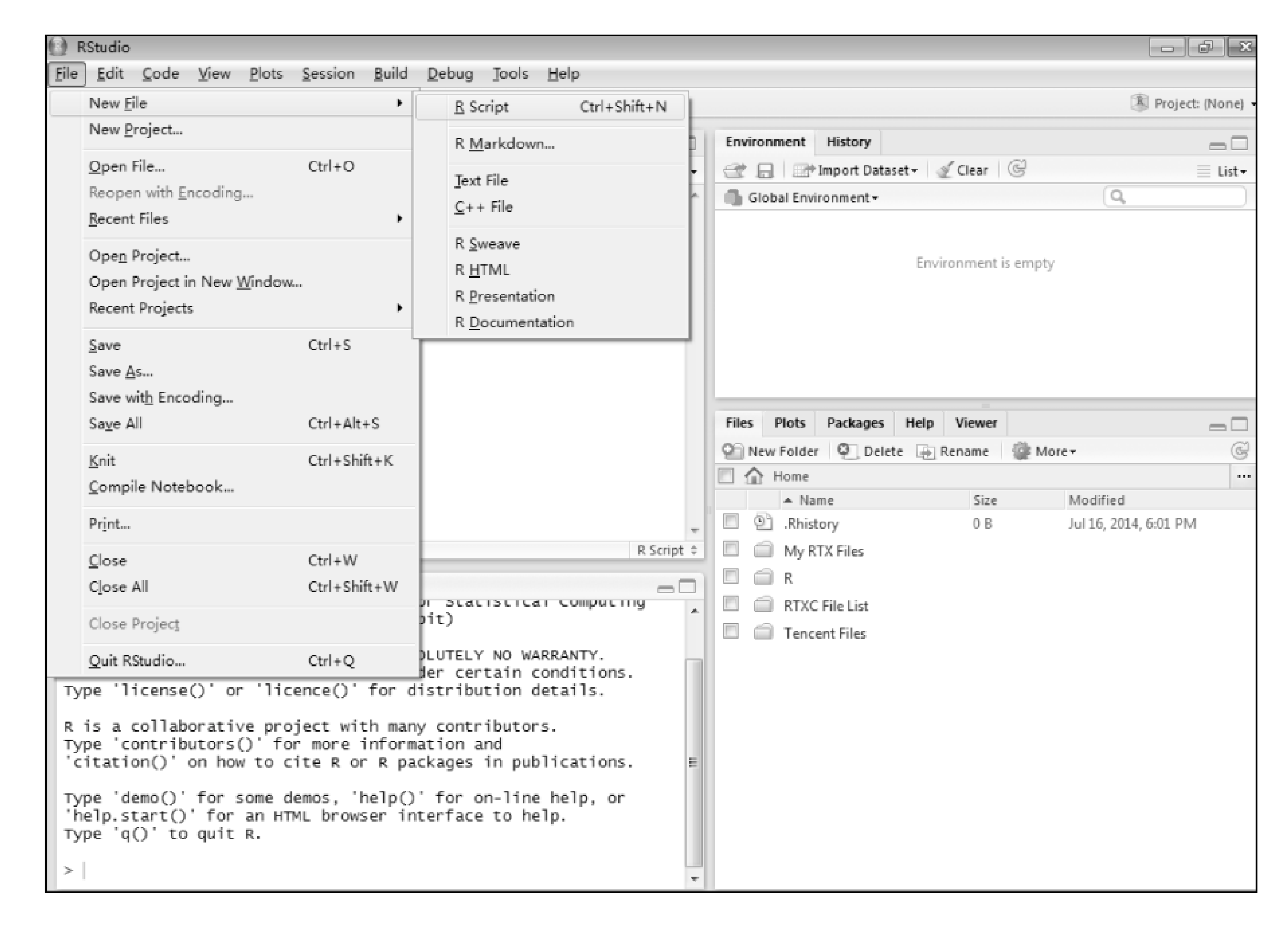


图 3.7 RStudio 文档创建

- Ctrl+Enter:运行光标所在行的 R 代码或者当前选中行的 R 代码。
- Ctrl+Shift+S:加载当前 R 文件并运行。
- Ctrl+D:删除整行。
- Ctr+Shift+C:注释/取消注释当前行。可以选中整个代码块进行注释。

3.2.3 R 的运行

Windows 中可单击 R 的快捷方式或在“开始”菜单中单击 R 软件图标运行 R。而在 Linux 系统下,需要在终端窗口中输入>R,然后按 Enter 键即可运行。在 Mac 上需要找到应用程序文件夹双击运行,如图 3.8 所示。

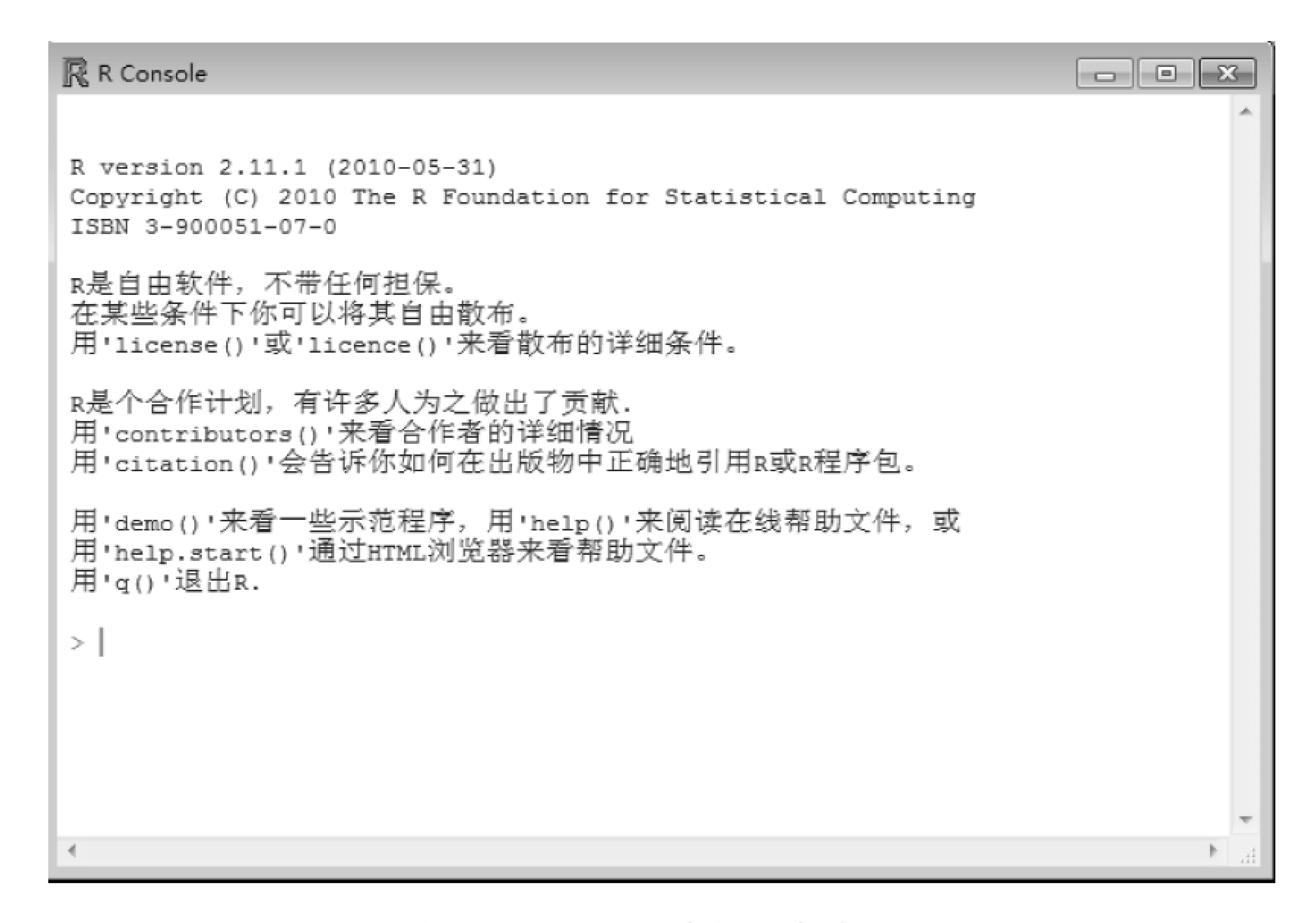


图 3.8 Windows 中的 R 启动界面

在软件使用过程中,遇到问题可尝试使用 R 中的帮助函数及文档。常用的帮助函数如表 3.1 所示。

表 3.1 R 常用帮助函数

函　数	功　能
help.start()	打开帮助文档首页(可查看入门和高级帮助手册、常见问题集)
data()	列出当前已加载包中所含的所有可用示例数据集
example("foo")	函数 foo 的使用示例(引号可以省略)
vignette("foo")	为主题 foo 显示指定的 vignette 文档
vignette()	列出当前已安装包中可用的 vignette 文档,一般为实用性介绍文章
help.search("foo")或?? foo	以 foo 为关键词搜索本地帮助文档
help(" foo")或? foo	查看函数 foo 的帮助(引号可以省略)
apropos("foo",mode="function")	列出名称中含有 foo 的所有可用函数
RSiteSearch("foo")	以 foo 为关键词搜索在线文档和邮件列表存档

使用 R 帮助文档有以下两种情况。

(1) 知道需要查询的关键字属于什么包，则在 console 中输入>? *** ，问号后为所需查询的关键字。

(2) 不知道该关键字属于什么包，则在 console 中输入>?? *** ，比第一种情况多出一个问号。

在查询关键字所属包的时候，若 R 中无此包，则需提前加载。以 ggplot 关键字为例，需要加载 ggplot2 包，需提前安装该包①：

```
> install.packages("ggplot2")
> library(ggplot2)
```

R 提供的大量帮助性功能，通过 R 常用函数及帮助文档查看某些函数如返回值或选项上的功能，可以帮助更好地学习编程，这也是它的亮点之一。

3.2.4 工作目录和工作空间

工作目录(Working Directory)是 R 用来读取文件、保存结果的默认目录。使用 getwd()命令可获得 R 的工作目录，使用 setwd()可重新设置当前的工作目录位置②，不过 setwd()重新设置的目录必须是已存在的目录位置，可以使用 dir.create()来创建新目录，然后通过 setwd()重新将工作目录指向新创建的目录。工作空间(Workspace)即 R 工作的环境，用户所定义的诸如向量、矩阵、函数、列表等对象就保存在工作空间中。

下面列出了获取和设定工作目录的常用方法。

1. 通过命令行获取和设定工作目录

```
> getwd()
[1] "C:/Users/min.li/Documents"
> setwd("E:/GTA 工作/R 软件/实务教材")
```

这里值得注意的是，在设置路径时，初学者经常会错误地使用>setwd("E:\GTA 工作\R 软件\实务教材")，但是可以使用命令> setwd("E:\\GTA 工作\\R 软件\\实务教材")。

注意：在使用 setwd()命令重新设置当前工作目录时，命令中的路径使用的是正斜杠(/)，即使在 Windows 系统下也是如此，而反斜杠在 R 中被作为一个转义符。

2. 通过工具栏获取和设定工作目录

如果是 R 语言，那么选择“文件”→“改变工作目录”命令，即可查看和设定工作目录，如图 3.9 所示。

如果是 RStudio，那么选择 Session→Set Working Directory 下的 To Source File Location 命令进行工作目录查看，选择 Choose Directory 命令进行工作目录的设定，如图 3.10 所示。

在一个 R 会话结束时，可以将当前工作空间保存到一个镜像中，并在下次启动 R 时自

---

① http://blog.sina.com.cn/s/blog_744c2fb701014su8.html.

② http://www.biostatistic.net/thread-3228-1-1.html.

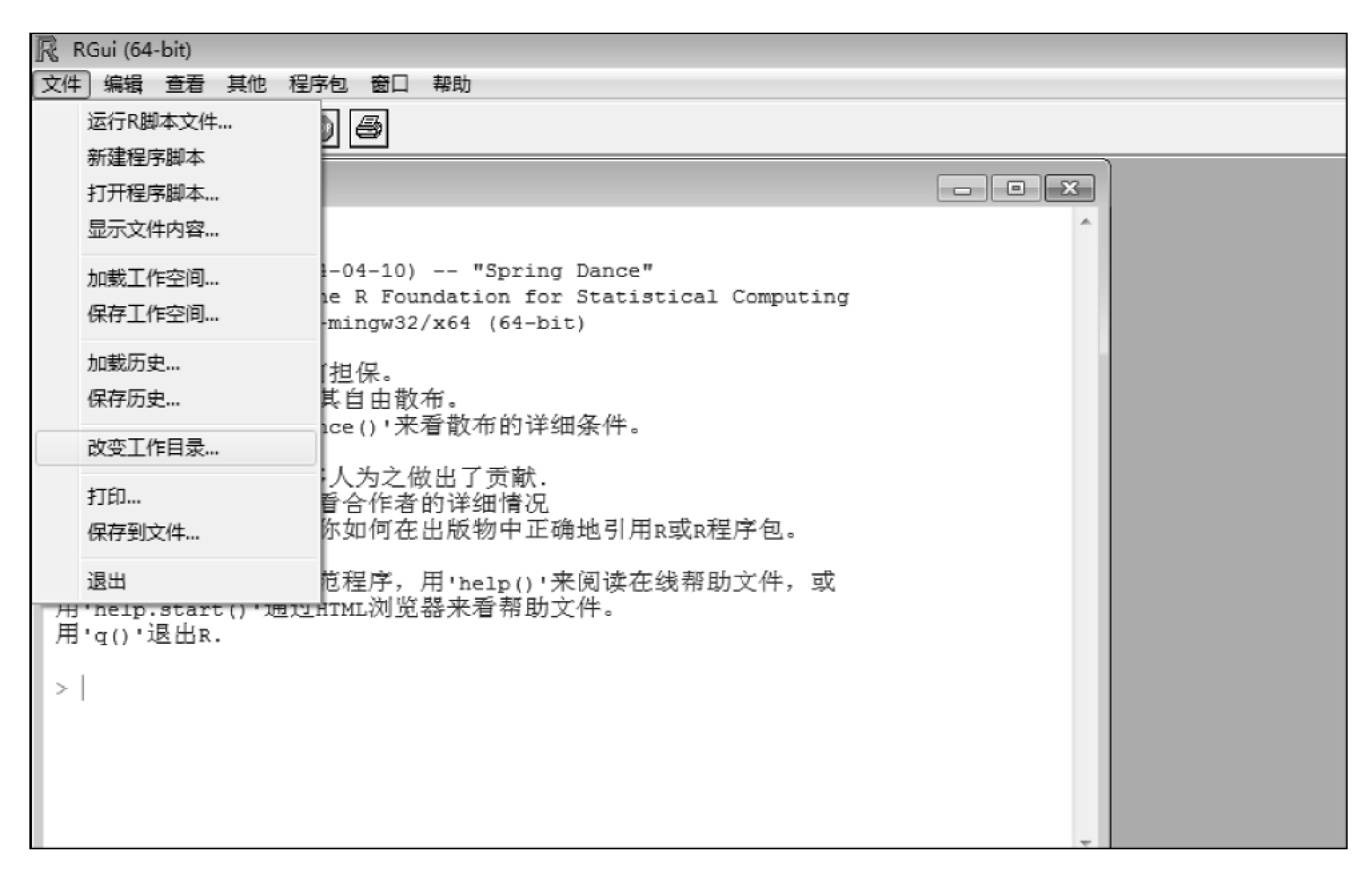


图 3.9　R 平台下的工作目录设定

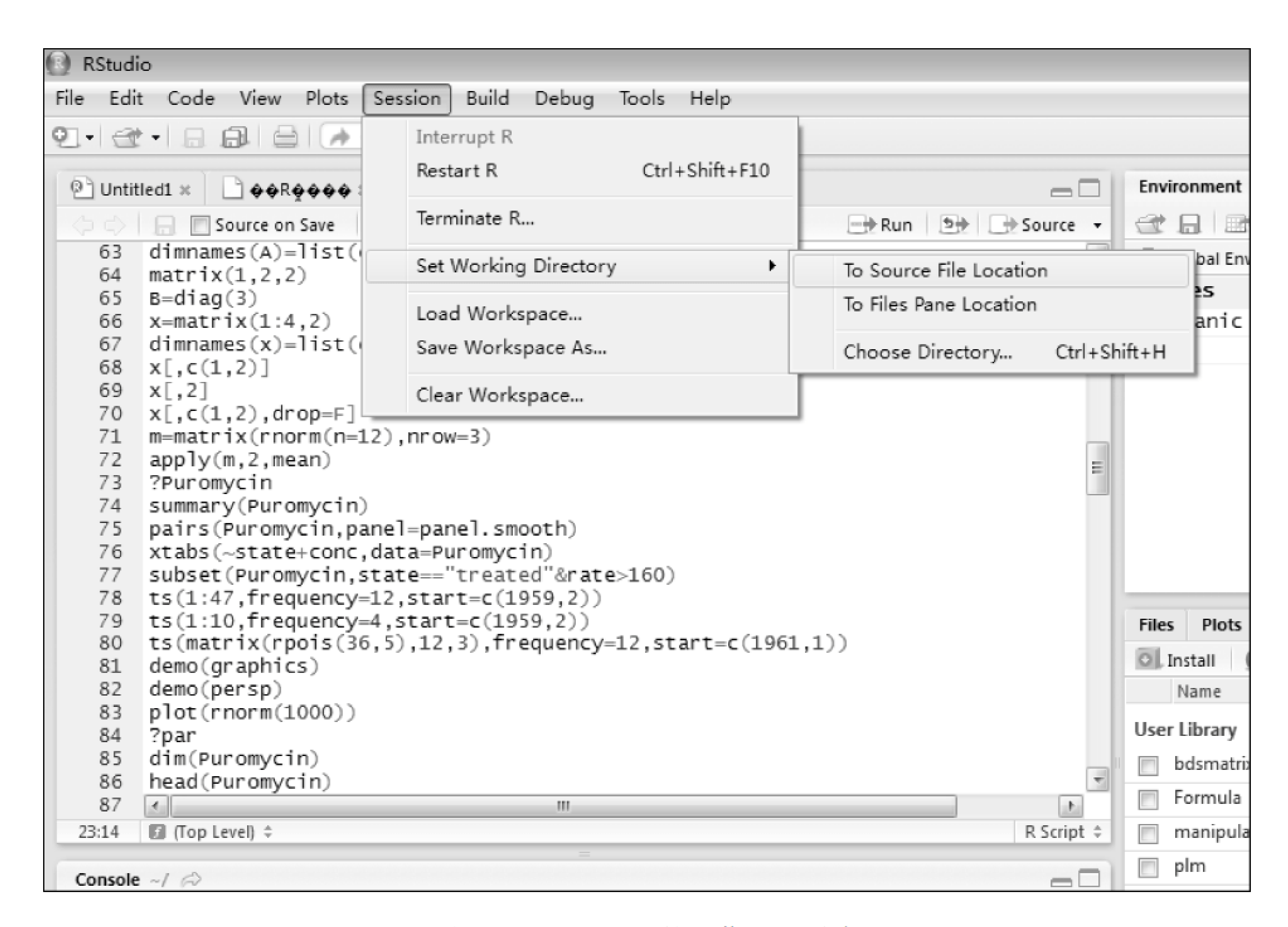


图 3.10　RStudio 的工作目录设定

动载入它。一般在关闭软件的时候会弹出提示窗口，如图 3.11 所示。

也可以使用命令语句在没有退出 R 软件的情况下保存工作空间，比如要去做其他事

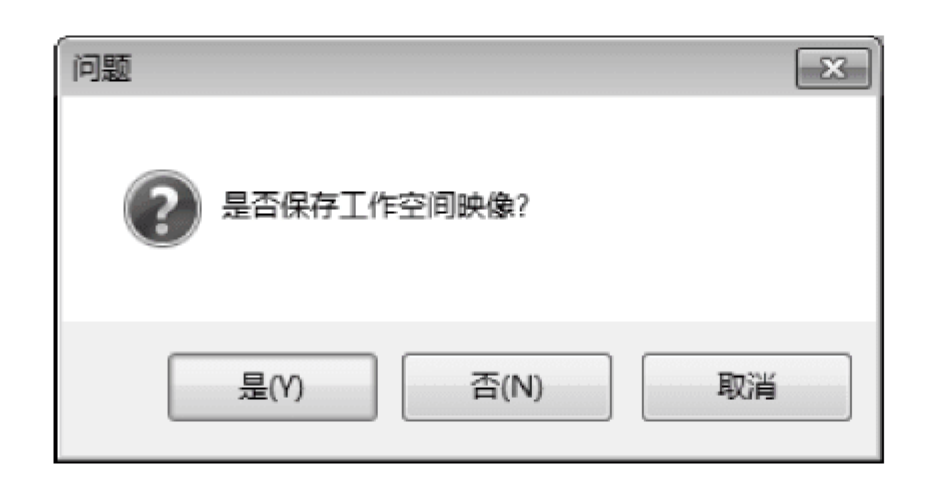


图 3.11　工作空间是否保留提示界面

情，防止中途偶然的电源或计算机故障导致数据丢失。

```
> save.image()
```

保存工作空间以后，下一次启动 R 时会自动还原。但是，工作空间不能保存当前打开的图形，退出以后就会消失，所以一定要记得保存制图代码。

在 R 工作空间的管理上涉及一些常用函数，如表 3.2 所示。

表 3.2　R 工作空间常用管理函数

函　数	功　能
getwd()	显示当前的工作目录
setwd("R-TEST")	重新设置当前的工作目录位置为 R-TEST
q()	退出 R
ls()	列出当前工作空间中的对象
rm(objectlist)	移除(删除)一个或多个对象
options()	显示或设置当前选项
help(options)	显示可用选项的说明
history(#)	显示最近使用过的#个命令(默认值为 25)
loadhistory("myfile")	载入一个命令历史文件(默认值为.Rhistory)
savehistory("myfile")	保存命令历史到文件 myfile 中(默认值为.Rhistory)
save.image("myfile")	保存工作空间到文件 myfile 中(默认值为.RData)
save(objectlist, file="myfile")	保存指定对象到一个文件中
load("myfile")	读取一个工作空间到当前会话中(默认值为.RData)

3.2.5　R 语言的帮助

单击 RStudio 图表展示窗口的 Help，这时展示窗口充当网页展示窗口。图 3.12 所示为帮助文件首页，里面展示的是已经安装到本地的帮助文档。如果使用 R 的话，也可以通过如下调用语句进入帮助，将弹出网页链接。

```
> help.start()
```

界面中最常使用的是 Reference(引用)部分的两个链接：

- Packages(R 软件包)：每个 Package 都有大量数据和可以读写修改的函数/程序，R 的强大也在于此，这里有来自全世界的统计学家和数据分析师编写的 R 软件包可以供用户使用，而 Packages 帮助文档包含 base 基础包和已经安装的包，单击软件包名

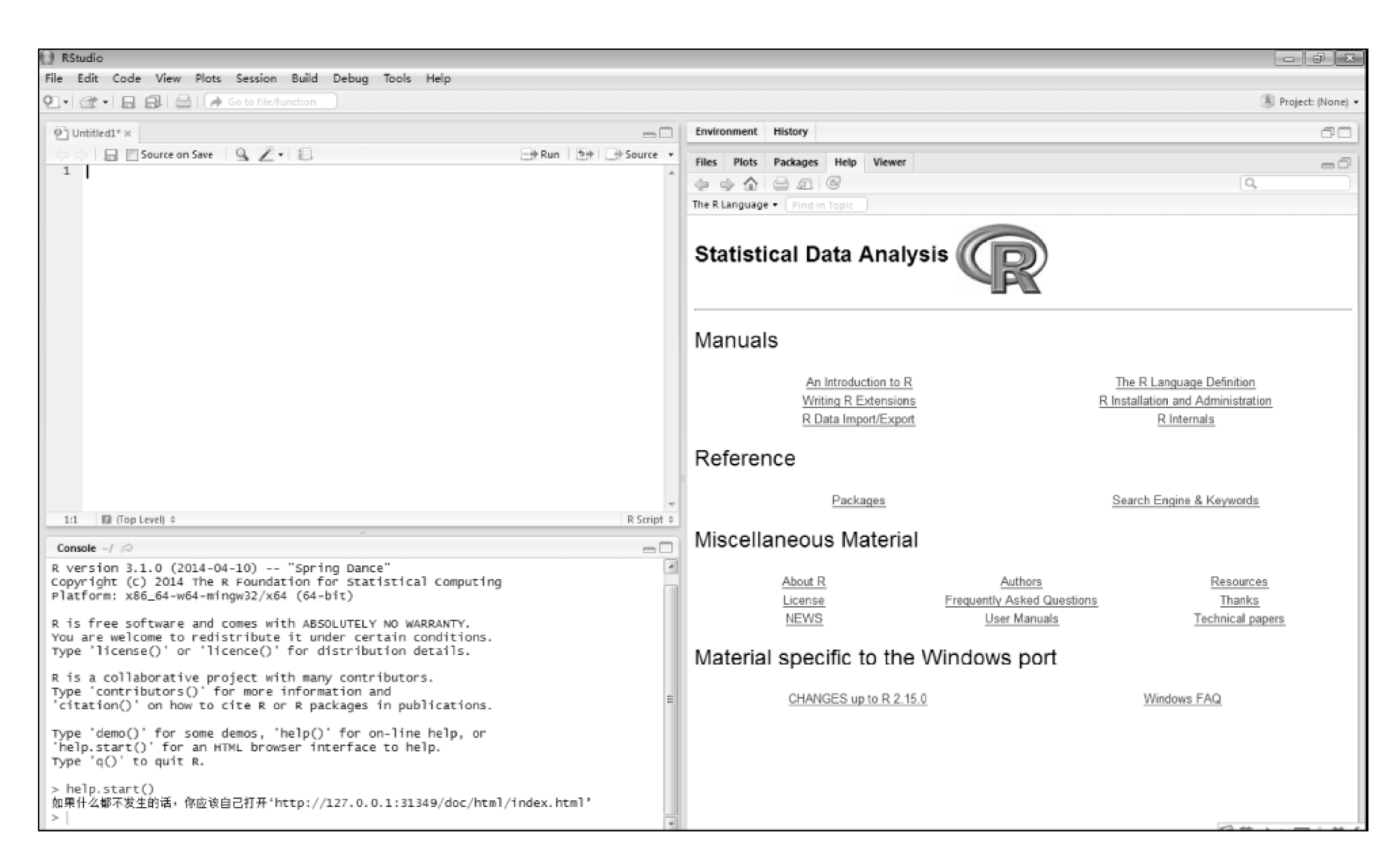


图 3.12　RStudio 的帮助界面

就可以查看函数和数据集。

- Search Engine & Keywords(搜索引擎与关键字)：输入关键字可以搜索相关的帮助文档。也可以通过比较快捷的方法查看函数帮助，例如：

查看某函数的帮助文档：

```
> help(function)
```

查看某函数的参数：

```
> args(function)
```

查看某函数的使用示例：

```
> example(function)
```

3.3　R 的包

3.3.1　包的获取

CRAN 上面发布了 5000 多个软件包，资源在哪？怎么使用呢？现在告诉读者怎么才能找到自己研究需要的包。

(1) 在 R 官网单击 CRAN，选择离自己距离近的镜像网，也可以直接单击网址 http://cran.rstudio.com/，进入图 3.13 所示页面。

(2) 在左侧导航条第一部分 CRAN 下可以单击 Task Views 链接查看任务视图，如图 3.14 所示。

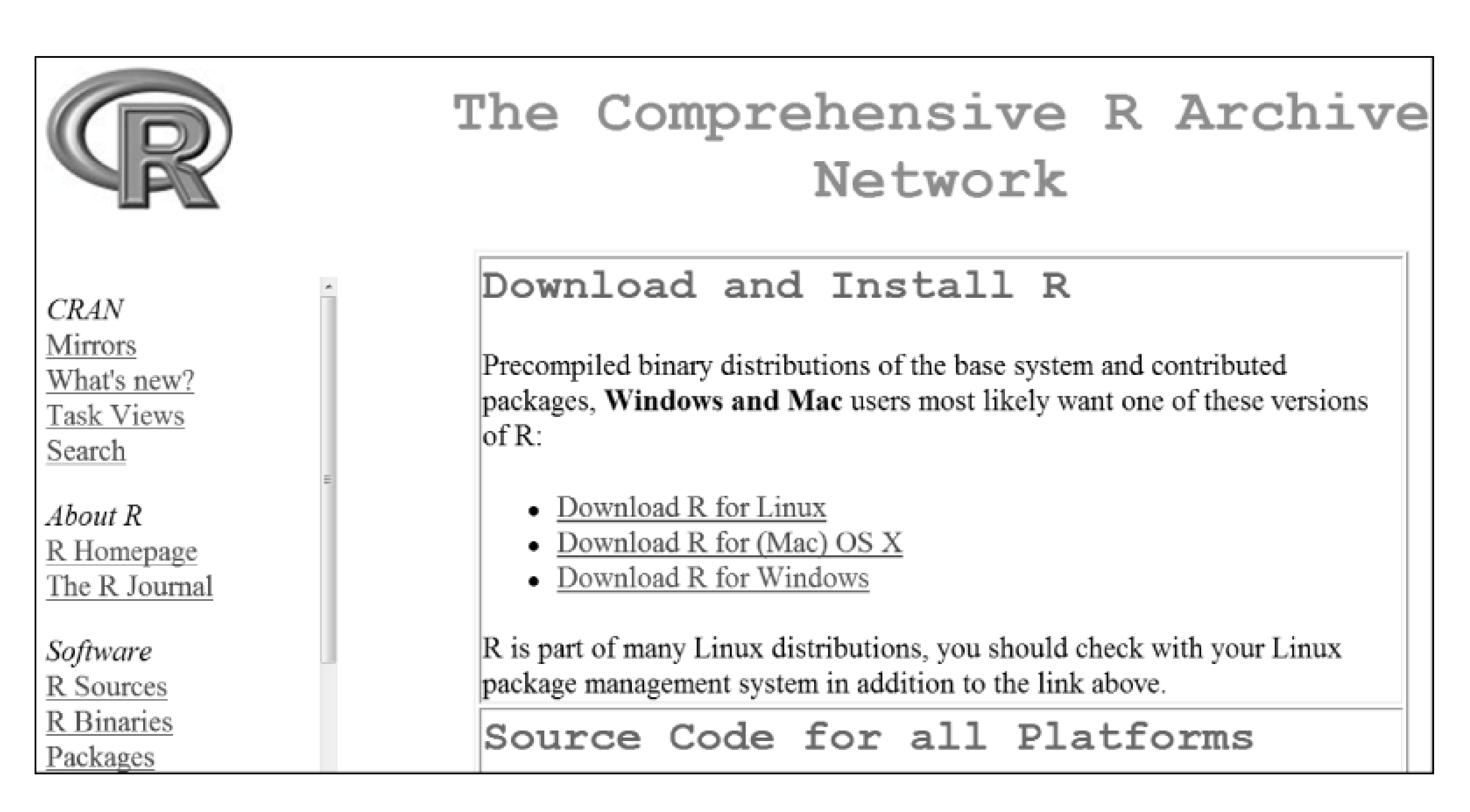


图 3.13　R 的资源获取界面

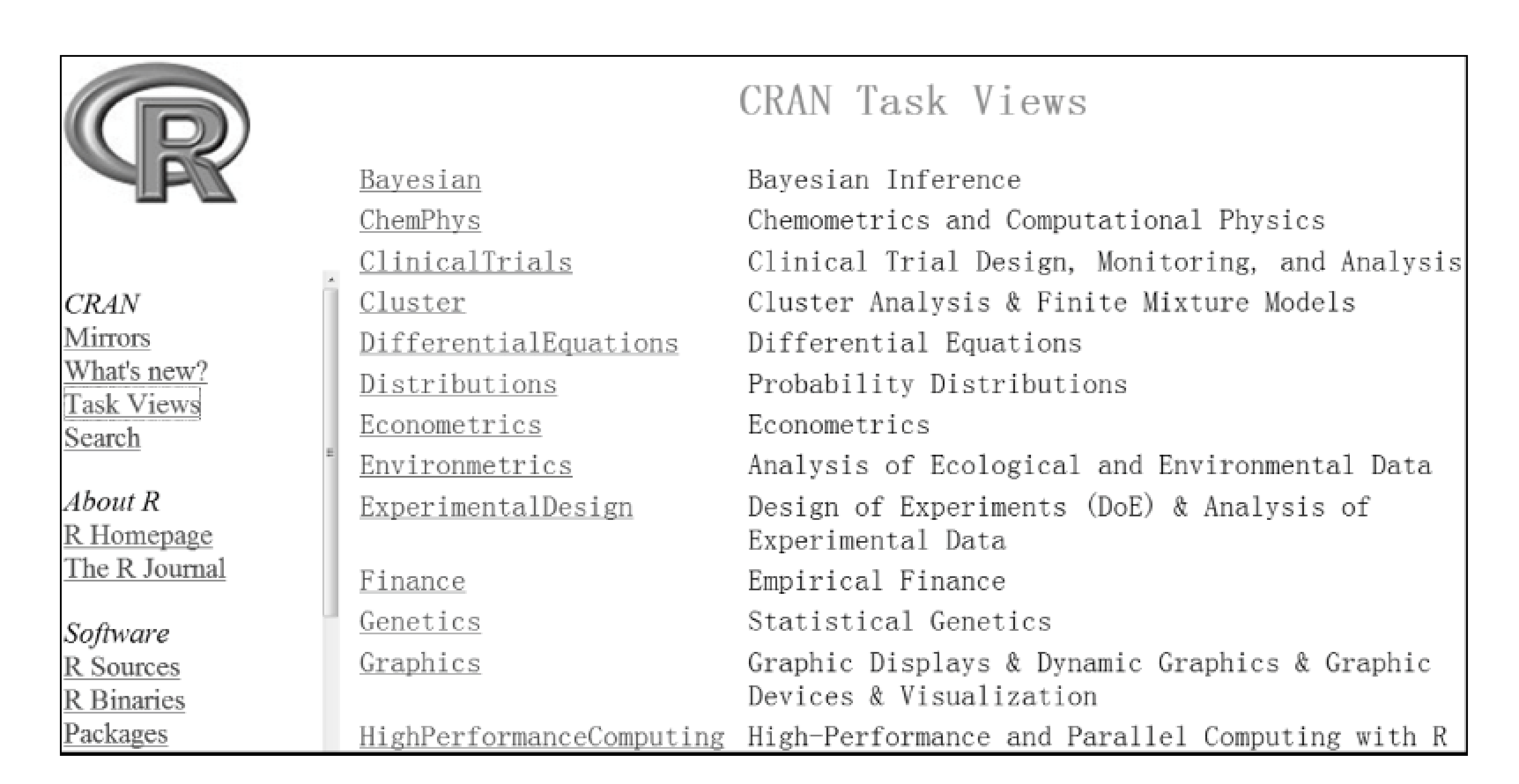


图 3.14　R 的包界面

Task Views 里面按照学科领域分门别类，现有的学科分类如表 3.3 所示。

表 3.3　R 应用领域

	CRAN Task Views	
Bayesian	Bayesian Inference	贝叶斯推理分析
ChemPhys	Chemometrics and Computational Physics	化学计量学和计算物理
ClinicalTrials	Clinical Trial Design，Monitoring，and Analysis	临床试验设计、监控和分析
Cluster	Cluster Analysis & Finite Mixture Models	聚类分析和有限混合模型
DifferentialEquations	Differential Equations	微分方程
Distributions	Probability Distributions	概率分布
Econometrics	Computational Econometrics	计量经济学

续表

	CRAN Task Views	
Environmetrics	Analysis of Ecological and Environmental Data	生态环境数据分析
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data	实验设计(DoE)和实验数据分析
Finance	Empirical Finance	实证金融
Genetics	Statistical Genetics	统计遗传学
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization	图形显示和动态图形和图形设备和可视化
HighPerformanceComputing	High-Performance and Parallel Computing with R	高性能计算和并行计算
MachineLearning	Machine Learning & Statistical Learning	机器学习
MedicalImaging	Medical Image Analysis	医学图像分析
MetaAnalysis	Meta-Analysis	荟萃分析
Multivariate	Multivariate Statistics	多元统计分析
NaturalLanguageProcessing	Natural Language Processing	自然语言处理
NumericalMathematics	Numerical Mathematics	计算数学
OfficialStatistics	Official Statistics & Survey Methodology	政府统计和社会调查
Optimization	Optimization and Mathematical Programming	最优化和数学规划(运筹学)
Pharmacokinetics	Analysis of Pharmacokinetic Data	药物动力学数据分析
Phylogenetics	Phylogenetics, Especially Comparative Methods	系统发生学,比较方法
Psychometrics	Psychometric Models and Methods	心理学模型和方法
ReproducibleResearch	Reproducible Research	可重复性研究
Robust	Robust Statistical Methods	稳健统计方法
SocialSciences	Statistics for the Social Sciences	社会科学统计
Spatial	Analysis of Spatial Data	空间数据分析
SpatioTemporal	Handling and Analyzing Spatio-Temporal Data	时空数据处理和分析
Survival	Survival Analysis	生存分析
TimeSeries	Time Series Analysis	时间序列分析
WebTechnologies	Web Technologies and Services	网络技术和服务
gR	gRaphical Models in R	制图模型

（3）单击相关学科，进入到该学科类别，以计量经济学（Econometrics）为例，如图 3.15 所示。

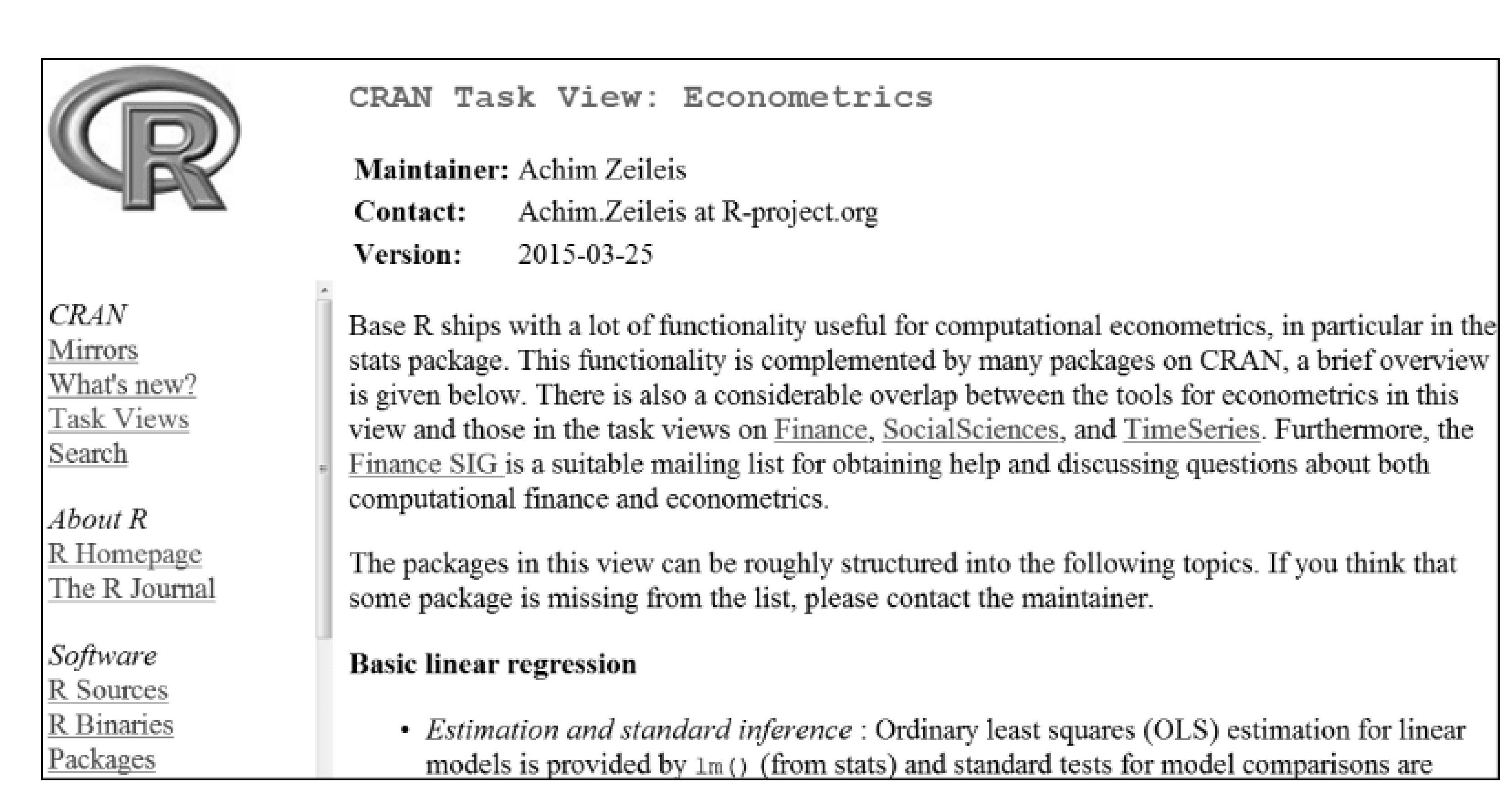


图 3.15　计量经济学的包文件示例

页面给出的这个计量经济学系列的包中大致包括以下几个主题。

- Linear regression models：线性回归模型。
- Micro econometrics：微观经济学。
- Further regression models：其他的回归模型。
- Basic time series infrastructure：基本的时间序列架构。
- Time series modeling：时间序列模型。

每个主题都进行了简单的介绍，包括各个主题下有些什么软件包，以及该软件包的功能。页面按照字母表顺序列出了该学科相关的所有 Packages，还给出了相关的 CRAN Task View，如果在这个 Task View 找不到，可以去相关的 Task View 继续找。有了导航就能轻松地找到需要的软件包了。

3.3.2　包的安装

找到需要的软件包 Name 以后就开始下载安装软件包。有以下几种方法可以进行安装：

（1）网页软件包名，以 plm（Linear Models for Panel Data）面板数据的线性模型软件包为例（如图 3.16 所示）来进行说明。

上面有作者及软件包的一些相关信息，选择 Windows 进行下载。

如果使用 R，选择“程序包”→“从本地 zip 文件安装程序包”命令，然后选择下载好的压缩包即可进行安装，如图 3.17 所示。

如果在 RStudio 里面进行安装，选择 Tools→Install Packages 命令，如图 3.18 所示。

弹出图 3.19 所示窗口，在 Install from 下拉列表中选择 Package Archive File 选项，然后选取之前下载的压缩包即可完成安装。

（2）知道自己需要下载安装的 Packages 的 name，可以直接在软件中完成下载安装。

如果使用 R，可以选择“程序包”→“安装程序包”命令进行安装，如图 3.20 所示。

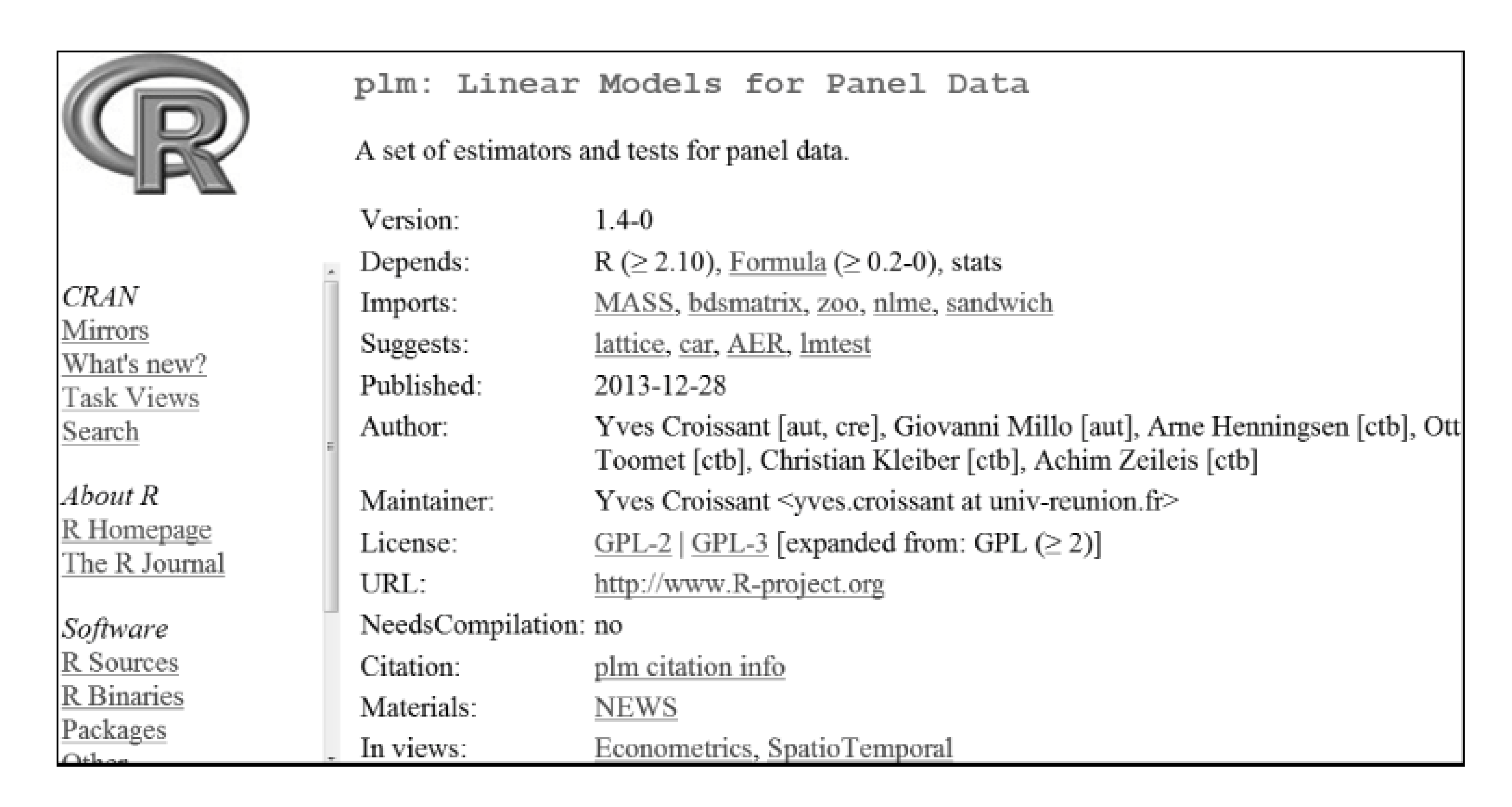


图 3.16 包的信息



图 3.17 本地程序包的安装

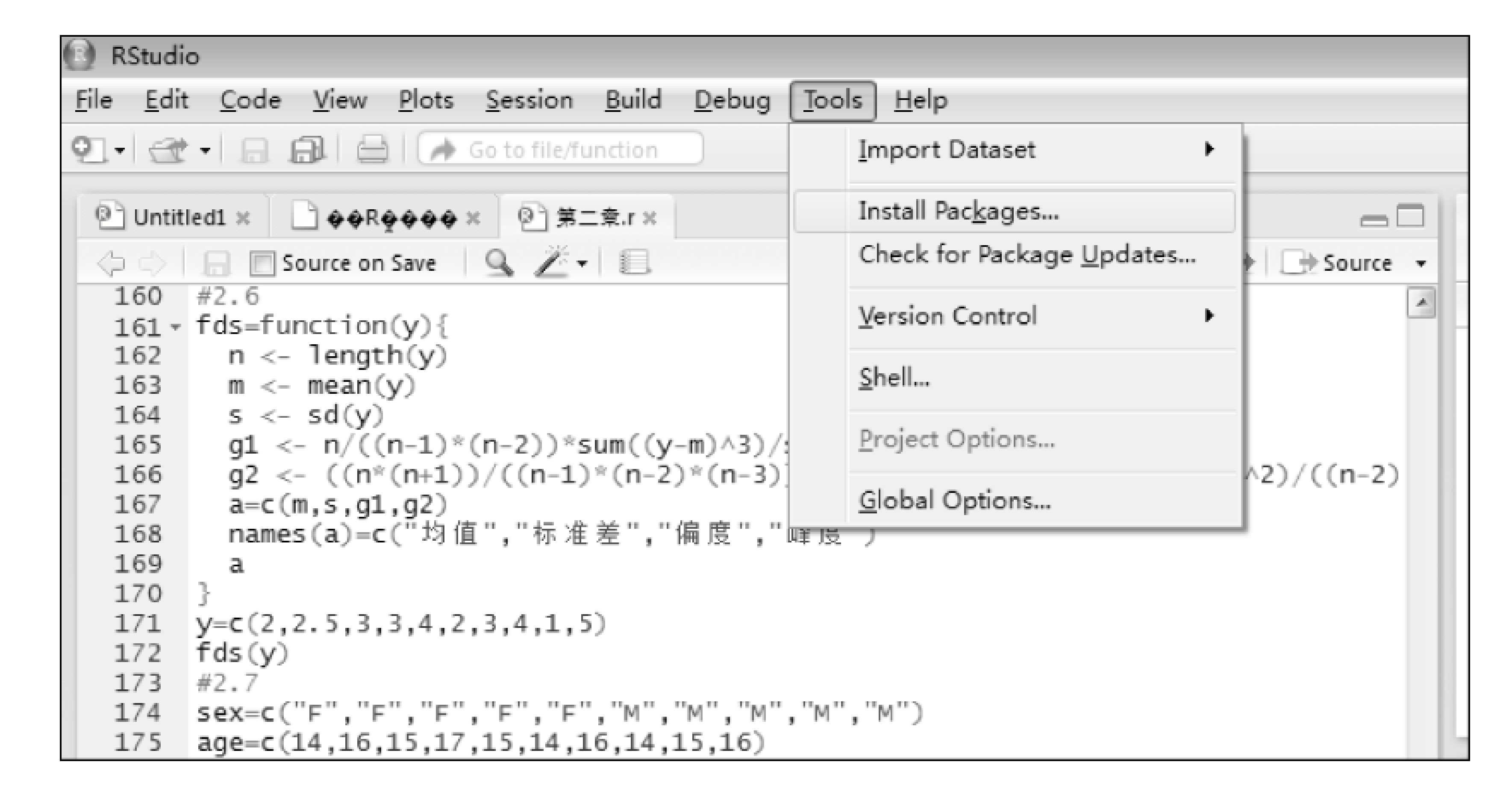


图 3.18 RStudio 包的菜单选择

Install Packages

Install from:

Package Archive File (.zip; .tar.gz)

Package archive:

Browse...

Install to Library:

C:/Users/wenhui.chen/Documents/R/win-library/3.1 [Default]

Install Cancel

图 3.19 RStudio 包的安装界面

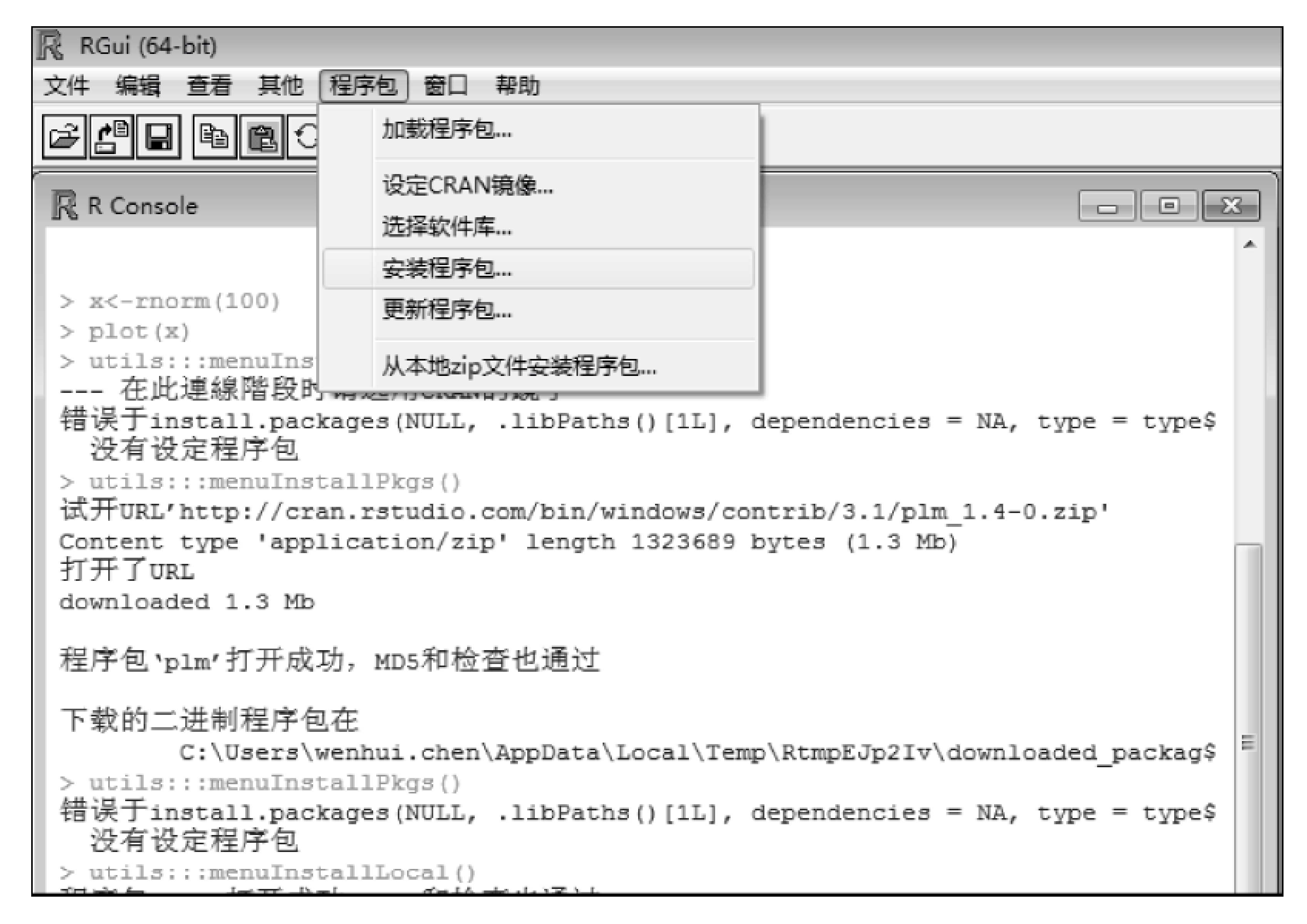


图 3.20 R 平台的包安装界面

选择后会弹出清单，清单是按照字母表顺序排列的，选择 plm 选项，单击“确定”按钮即可完成安装，如图 3.21 所示。

如果使用 RStudio，可以单击右下方小窗口的 Packages，如图 3.22 所示。

显示的是已经安装好的软件包，单击 Update 按钮可以实现对已经安装 Packages 的升级。单击 Install 按钮会弹出图 3.23 所示窗口。

在 Install from 下拉列表中选择 Repository(CRAN，CRANextra)选项，表示从网上下载需要安装的软件包，安装路径 Library 是默认的。在 Rackages(separate maltiple with space or comma)文本框中输入需要下载安装的 Packages 的名称 plm，单击 Install 按钮就可以进行安装了。安装好的 Packages 会进入软件包库 Library 里面，并且自动显示在已安装的条目下。

(3) 无论 R 还是 RStudio 都可以用命令实现安装，语句如下：

```
install.packages("plm")
```

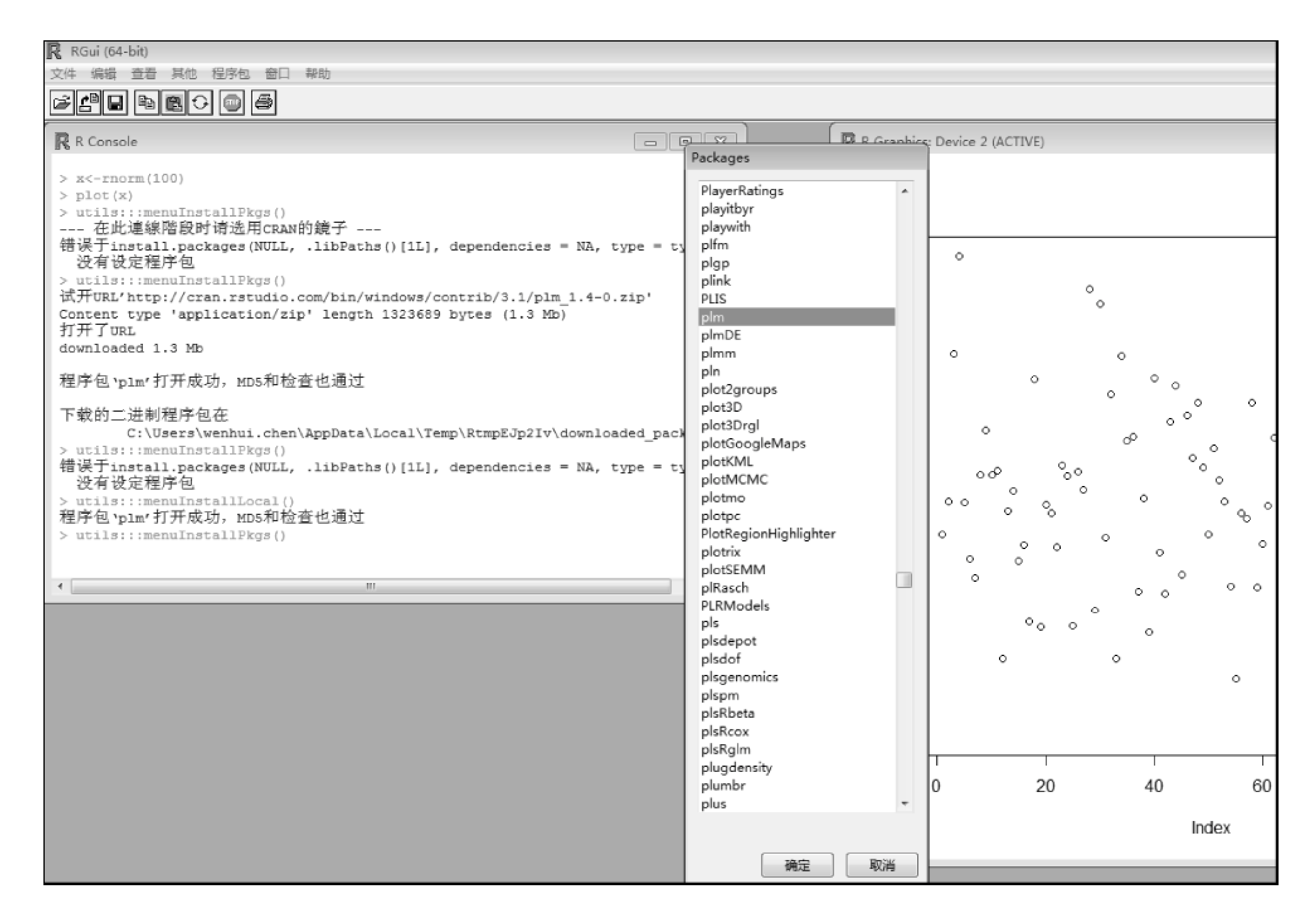


图 3.21 R 平台的包名称选择

Files | Plots | Packages | Help | Viewer

Install | Update

	Name	Description	Version
User Library			
☐	manipulate	Interactive Plots for RStudio	0.98.945
☐	rstudio	Tools and Utilities for RStudio	0.98.945
System Library			
☐	boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-11
☐	class	Functions for Classification	7.3-10
☐	cluster	Cluster Analysis Extended Rousseeuw et al.	1.15.2
☐	codetools	Code Analysis Tools for R	0.2-8
☐	compiler	The R Compiler Package	3.1.0
☑	datasets	The R Datasets Package	3.1.0
☐	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...	0.8-61
☑	graphics	The R Graphics Package	3.1.0
☑	grDevices	The R Graphics Devices and Support for Colours and Fonts	3.1.0
☐	grid	The Grid Graphics Package	3.1.0
☐	KernSmooth	Functions for kernel smoothing for Wand & Jones (1995)	2.23-12
☐	lattice	Lattice Graphics	0.20-29
☐	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-31

图 3.22 显示 RStudio 中已经安装好的包

建议读者先在官网了解软件包的功能，然后再在软件中直接进行下载安装。

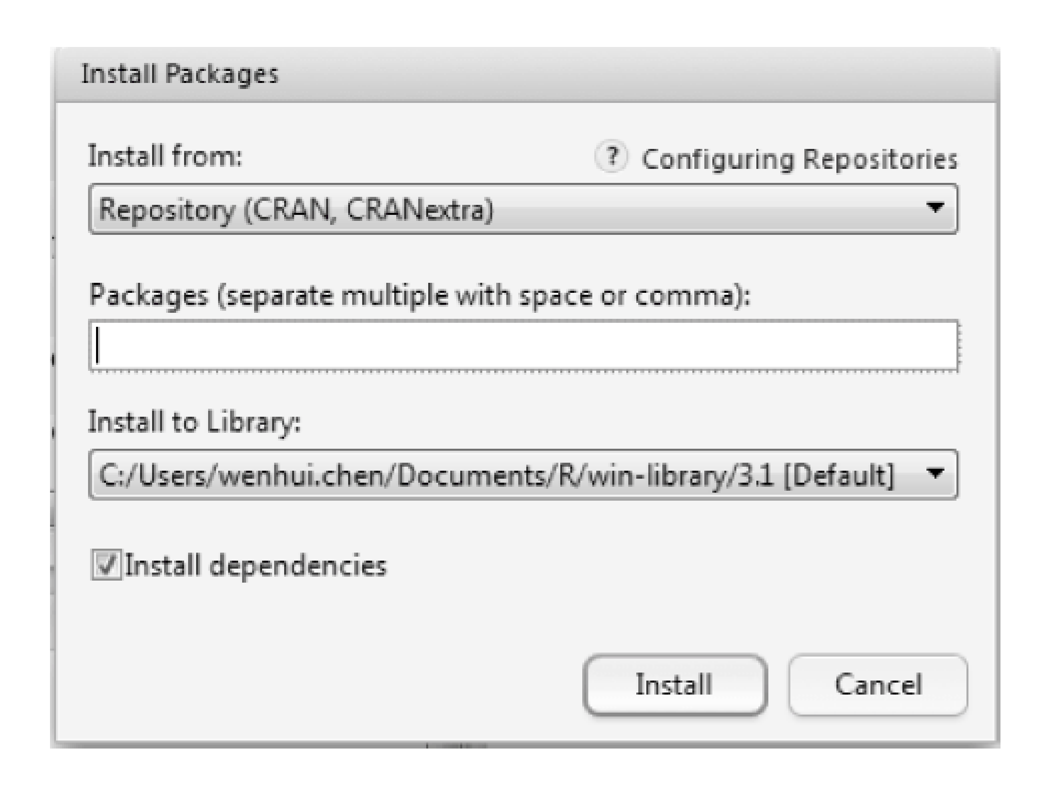


图 3.23　RStudio 包的安装界面

3.3.3　包的加载

Packages 安装好以后要加载才能使用，没有加载的话软件包中的函数是无法调用的。R 开启后自带的标准包已经加载好，可以直接使用，比如 base、datasets、graphics 等。不带任何参数的 library()打开当前系统中所有包介绍信息

```
>library()
```

如果要使用其他软件包，可以使用命令语句 library()进行加载，使用以后将它从内存释放，例如：

```
#加载 MASS 软件包
>library(MASS)
#卸载 MASS 软件包
>detach("package:MASS", unload = TRUE)
```

如果使用 RStudio，可以用更加快捷的方式进行 Packages 的加载，如图 3.24 所示。RStudio 右下方小窗口的 Packages 选项，菜单中显示了已经安装好的软件包，选中该软件包可以实现加载，取消选中可以使软件包从内存中释放。

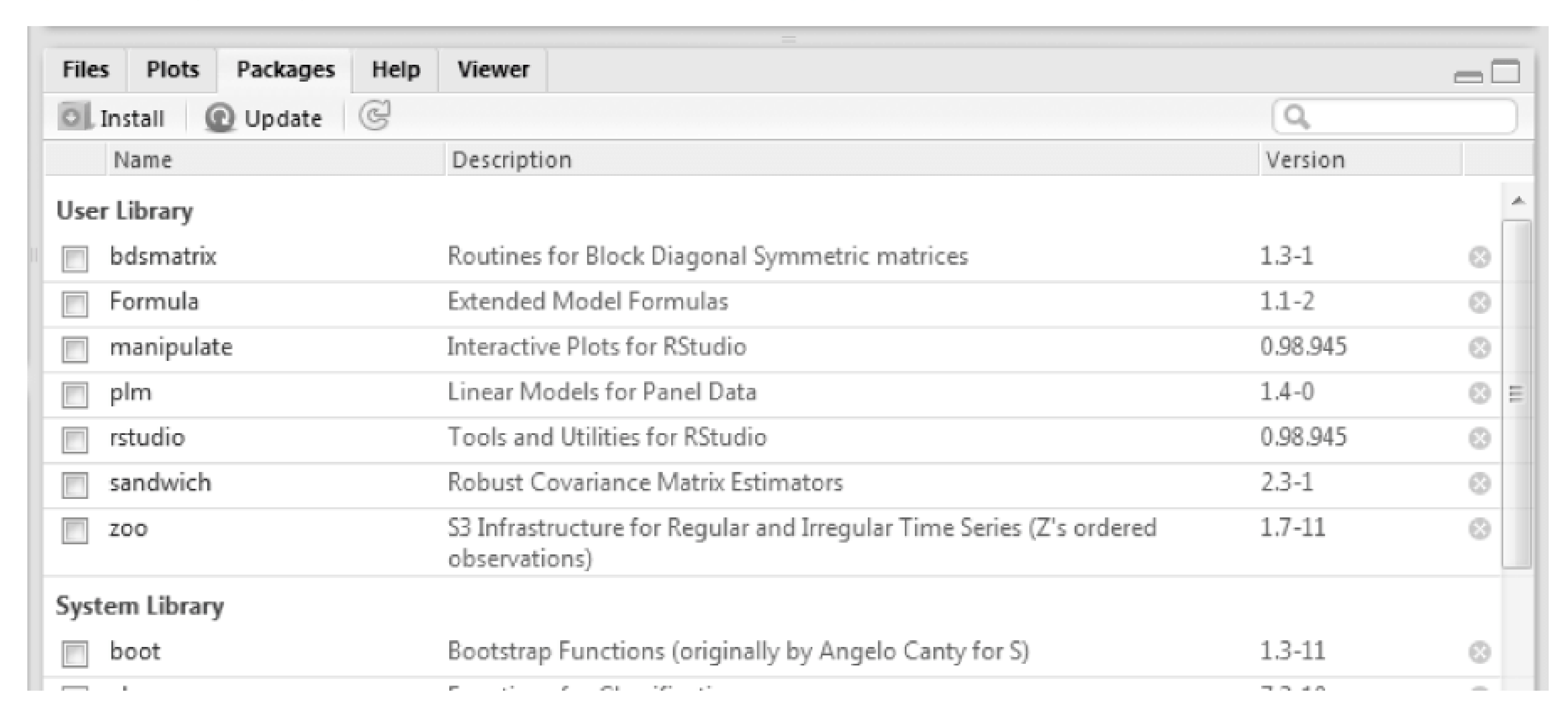


图 3.24　通过界面操作加载包

每个 library 都有许多数据，可以使用 data()查看 library 中的数据。例如，调出数据 Titanic。

```
> data(Titanic)
> Titanic
```

运行结果：

```
,, Age = Child, Survived = No
      Sex
Class  Male Female
  1st     0    0
  2nd     0    0
  3rd    35   17
  Crew    0    0

,, Age = Adult, Survived = No
      Sex
Class  Male Female
  1st   118    4
  2nd   154   13
  3rd   387   89
  Crew  670    3

,, Age = Child, Survived = Yes
      Sex
Class  Male Female
  1st     5    1
  2nd    11   13
  3rd    13   14
  Crew    0    0

,, Age = Adult, Survived = Yes
      Sex
Class  Male Female
  1st    57  140
  2nd    14   80
  3rd    75   76
  Crew  192   20
```

3.3.4　包的使用

成功载入包后便可以调用包中相应的函数及数据集。R 包一般都包含了相应的数据集及示例代码，方便用户了解该包的功能及使用。包中函数描述及数据集的信息包含在帮助系统中，可以运用 help()查看包的功能及包中函数和数据集的使用等具体细节。

CHAPTER 4

第4章 R语言基本操作

R 语言有一些基础的操作命令，包括赋值、向量运算、矩阵运算、元素运算、逻辑运算及简单的函数运算等。需要注意的是，R 语言区分字母的大小写，比如变量 A 和变量 a 代表不同的变量名称。

4.1 数据结构

R 的数据结构包括向量、矩阵、数组、数据框、列表和因子等。R 可以处理的数据类型（模式）包括数值型、字符型、逻辑型（TRUE/FALSE）、复数型（虚数）和原生型（字节），如图 4.1 所示。

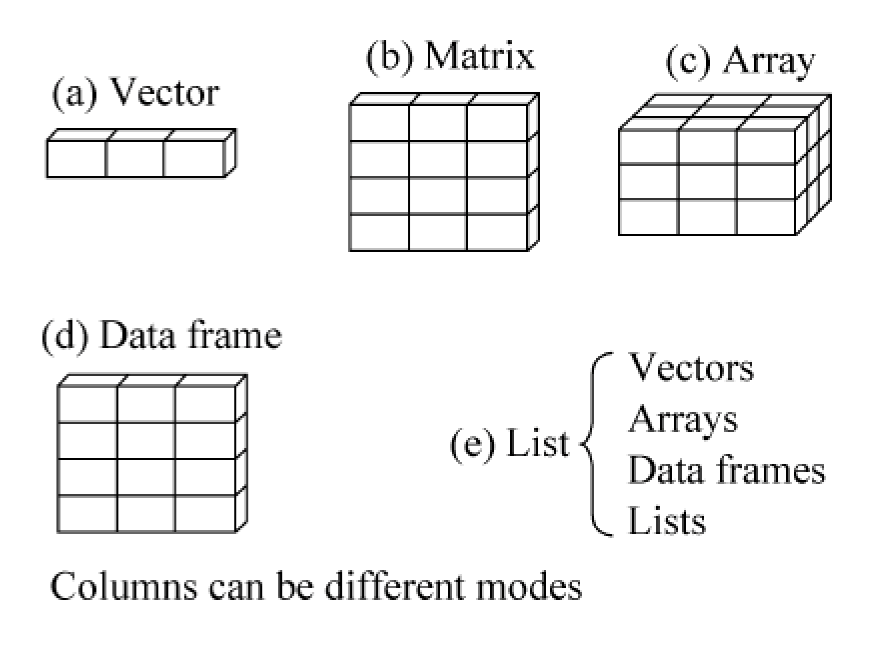


图 4.1 数据类型结构图

1. 向量

向量 Vector(1 * n，n * 1)是用于存储数值型、字符型或逻辑型数据的一维数组。其中，只含一个元素的向量称为标量。根据向量中元素类型的不同，可将向量分为数值型向量、字符型向量等。

2. 矩阵

矩阵 Matrix(n * m)是一个二维数组，其中的每个元素是相同的数据类型，比如数值型、

字符型等。

3. 数组

数组 Array(n * m * l)类似于矩阵，与矩阵不同的是，其维度可以大于 2。

4. 数据框

数据框 Data frame：(n * m)是 R 中最常处理的数据结构，其最大的特点是不同的列可以包含不同的数据类型。当数据有多种数据类型时，使用数据框可以将数据集放入一个矩阵。

5. 列表

列表(List)是一些对象的有序集合，其中的对象可以是任何的数据结构类型，比如向量、矩阵、数据框等。列表由向量派生而来，是 R 中最复杂的一种数据类型。

6. 因子

因子(Factor)是 R 定义的一种特殊的数据类型。因子指的是名义型变量或有序型变量，例如(类型 1、类型 2、类型 3)为名义型变量，而(优、良、中、差)是有序型变量。

4.2　数据的基本操作

本章主要介绍 R 中一些基本的数据操作，包括赋值、创建、运算及数据的导入等操作。

4.2.1　赋值和创建

1. 赋值操作

R 语句由函数和赋值构成。R 使用<－(像一个小箭头)表示赋值给箭头指向的变量，也可用传统的＝作为赋值符号。例如下面语句：

```
> x<- 10
> x
[1] 10
> 9 -> y
> y
[1] 9
> a<- 10 -> b
> a
[1] 10
> b
[1] 10
```

结果输出表示为向量形式，前面的[1]表示这是向量的第一个元素。另外，使用 R 可以反转赋值方向，小箭头指向的方向被赋予一定的函数值。当然，R 允许使用＝为对象赋值，但是这样写的 R 程序并不多，因为它不是标准语法，某些情况下用等号赋值会出现问题。

避免使用关键字作为变量名，如 c<－ 10，这样的命名方式在计算中极易报错，其原因在于 c()是一个连接函数。

2. 数据的创建

1）向量的建立

函数 c()可用来创建向量,各类向量如下所示：

```
> x <- c(1,2,3)
> y <- c("国","泰","安")
> z <- c(T,T,F)
```

其中,x 是数值型向量；y 是字符型向量；z 是逻辑型向量。

通过在方括号中给定元素所处位置的数值,可以访问向量中的元素。例如访问向量 x 中的第二个和第三个元素。

```
> x[c(2, 3)]
[1] 2 3
```

单个向量中的数据必须拥有相同的类型或模式(数值型、字符型或逻辑型),同一向量中不能混杂不同模式的数据。下面对不同的数据类型进行举例。

(1) 数值型向量建立。

统计分析中最常用的是数值型的向量,可以使用以下 4 种函数进行数值型向量建立。

① seq()或“:”(若向量具有较为简单的规律)

seq(from = 1, to = 1, by =步长, length. out =序列长度)

```
> 1:10
[1]  1  2  3  4  5  6  7  8  9  10
> 1:10 - 1
[1] 0 1 2 3 4 5 6 7 8 9
> 1:(10 - 1)
[1] 1 2 3 4 5 6 7 8 9
> z <- seq(1,5,by = 0.5)
> z
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> z <- seq(1,10,length = 11)
> z
[1]  1.0  1.9  2.8  3.7  4.6  5.5  6.4  7.3  8.2  9.1  10.0
```

② rep( )(若向量具有较复杂的规律)

rep(x, times =序列循环次数, length. out =序列长度, each =每个元素出现次数)

```
> z <- rep(2:5,2)
> z
[1] 2 3 4 5 2 3 4 5
> z <- rep(2:5,rep(2,4))
> z
[1] 2 2 3 3 4 4 5 5
> z <- rep(1:3,times = 4,each = 2)
> z
[1] 1 1 2 2 3 3 1 1 2 2 3 3 1 1 2 2 3 3 1 1 2 2 3 3
```

③ c( )(若向量没有什么规律)

前面已经给出了例子,在此不再赘述。

④ scan( )(通过键盘逐个输入)

```
> z <- scan()
1: 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
10:
Read 9 items
> z
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> z <- sequence(3:5)
> z
[1] 1 2 3 1 2 3 4 1 2 3 4 5
> z <- sequence(c(10,5))
> z
[1]  1  2  3  4  5  6  7  8  9  10  1  2  3  4  5
```

(2) 字符型向量建立。

字符型向量也是经常用到的,比如图表的标签。字符串输入时可以使用单引号(''),也可以使用双引号(" "),有以下两种常用函数进行字符型向量建立。

① c( )

前面已经举过例子,这里就不再举例。

② paste()

paste (…, sep = " ", collapse = NULL)

```
> labs <- paste(c("X",'Y'),1:10,sep = "")
> labs
[1] "X1" "Y2" "X3" "Y4" "X5" "Y6" "X7" "Y8" "X9" "Y10"
```

2) 矩阵的建立

可通过函数 matrix 创建矩阵。一般使用格式为:

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
```

其中,data 包含了矩阵的元素,nrow 和 ncol 用以指定行和列的维数,byrow 则表明矩阵应当按行填充(byrow=TRUE)还是按列填充(byrow=FALSE),默认情况下按列填充。dimnames 包含了可选的、以字符型向量表示的行名和列名。例如:

```
#按行排列建立 3×3 的矩阵
> a <- matrix(1:9,3,3,T)
> a
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
#按列排列建立 3×3 的矩阵
```

```
> b<-matrix(1:9,3,3,F)
> b
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> cells<-c(9,0,4,6)
> rnames<-c("R1","R2")
> cnames<-c("C1","C2")
> mymatrix<-matrix(cells,2,2,T,dimnames=list(rnames,cnames))
> mymatrix
   C1 C2
R1  9  0
R2  4  6
> mymatrix<-matrix(cells,2,2,F,dimnames=list(rnames,cnames))
> mymatrix
   C1 C2
R1  9  4
R2  0  6
```

可以使用下标和方括号[ ]的方式来选择和提取矩阵中的行、列元素。例如，X[i，]是指提取矩阵 X 中第 i 行的所有元素，X[，j]是指提取矩阵 X 中第 j 列的所有元素，X[i，j]是指提取矩阵 X 中第 i 行、第 j 列的元素。选择多行或多列时，下标 i 和 j 可为数值型向量。

```
> x<-matrix(1:10,nrow=2)
> x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
> x[,2]
[1] 3 4
> x[2,]
[1]  2  4  6  8  10
> x[2,3]
[1] 6
> x[1,c(3,5)]
[1] 5 9
```

3）数组的建立

数组可通过 array 函数创建，形式如下：

```
array ( vector, dimension, dimnames )
```

其中，vector 包含了数组中的数据元素，dimension 是一个数值型向量，给出了各个维度下标的最大值，dimnames 是可选的、各维度名称标签的列表。下面给出一个创建三维(2×3×4)数值型数组的示例。

```
> dim1 <- c("A1","A2")
> dim2 <- c("B1","B2","B3")
> dim3 <- c("C1","C2","C3","C4")
> z <- array(1:24,c(2,3,4),dimnames = list(dim1,dim2,dim3))
> z
,, C1
    B1  B2  B3
A1   1   3   5
A2   2   4   6
,, C2
    B1  B2  B3
A1   7   9  11
A2   8  10  12
,, C3
    B1  B2  B3
A1  13  15  17
A2  14  16  18
,, C4
    B1  B2  B3
A1  19  21  23
A2  20  22  24
```

数组是矩阵的一种推广,在编写新的统计方法时可能很有用。从数组中选取元素的方式与矩阵相同,例如:

```
> z[1,2,3]
[1] 15
```

4) 数据框的建立

数据框可通过函数 data.frame()创建:

```
data.frame (col1, col2, col3)
```

其中,列向量 col1, col2, col3,…可为任何类型(如字符型、数值型或逻辑型),每一列的名称可由函数 names 指定。下面举例说明。

表 4.1 中包含了不同类型的数据,通过 R 语言创建此数据框。

表 4.1 病人信息登记表

病人编号 (PatientID)	入院时间 (AdmDate)	年龄 (Age)	糖尿病类型 (Diabetes)	病情 (Status)
1	10/15/2009	25	Type1	Poor
2	11/01/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

```
> patientID <- c(1,2,3,4)
> age <- c(25,34,28,52)
> diabetes <- c("Type1","Type2","Type1","Type1")
> status <- c("Poor","Improved","Excellent","Poor")
> patientdata <- data.frame(patientID,age,diabetes,status)
> patientdata
  patientID age diabetes status
1         1  25    Type1      Poor
2         2  34    Type2      Improved
3         3  28    Type1      Excellent
4         4  52    Type1      Poor
> summary(patientdata)
   patientID       age          diabetes       status
Min.    :1.00   Min.   :25.00   Type1:3    Excellent :1
1st Qu. :1.75   1st Qu.:27.25   Type2:1    Improved  :1
Median  :2.50   Median :31.00                Poor      :2
Mean    :2.50   Mean   :34.75
3rd Qu. :3.25   3rd Qu.:38.50
Max.    :4.00   Max.   :52.00
```

选取数据框中元素的方式有以下几种，以上述案例为例。

```
> patientdata[1:3]
  patientID age diabetes
1         1  25    Type1
2         2  34    Type2
3         3  28    Type1
4         4  52    Type1
> patientdata[c("age","status")]
   age  status
1  25   Poor
2  34   Improved
3  28   Excellent
4  52   Poor
> patientdata $ diabetes
[1] Type1 Type2 Type1 Type1
Levels: Type1 Type2
```

可以通过以下代码语句查看及修改数据框的数据(如图 4.2 所示)。

```
> data.entry(patientdata)
> edit(patientdata)
```

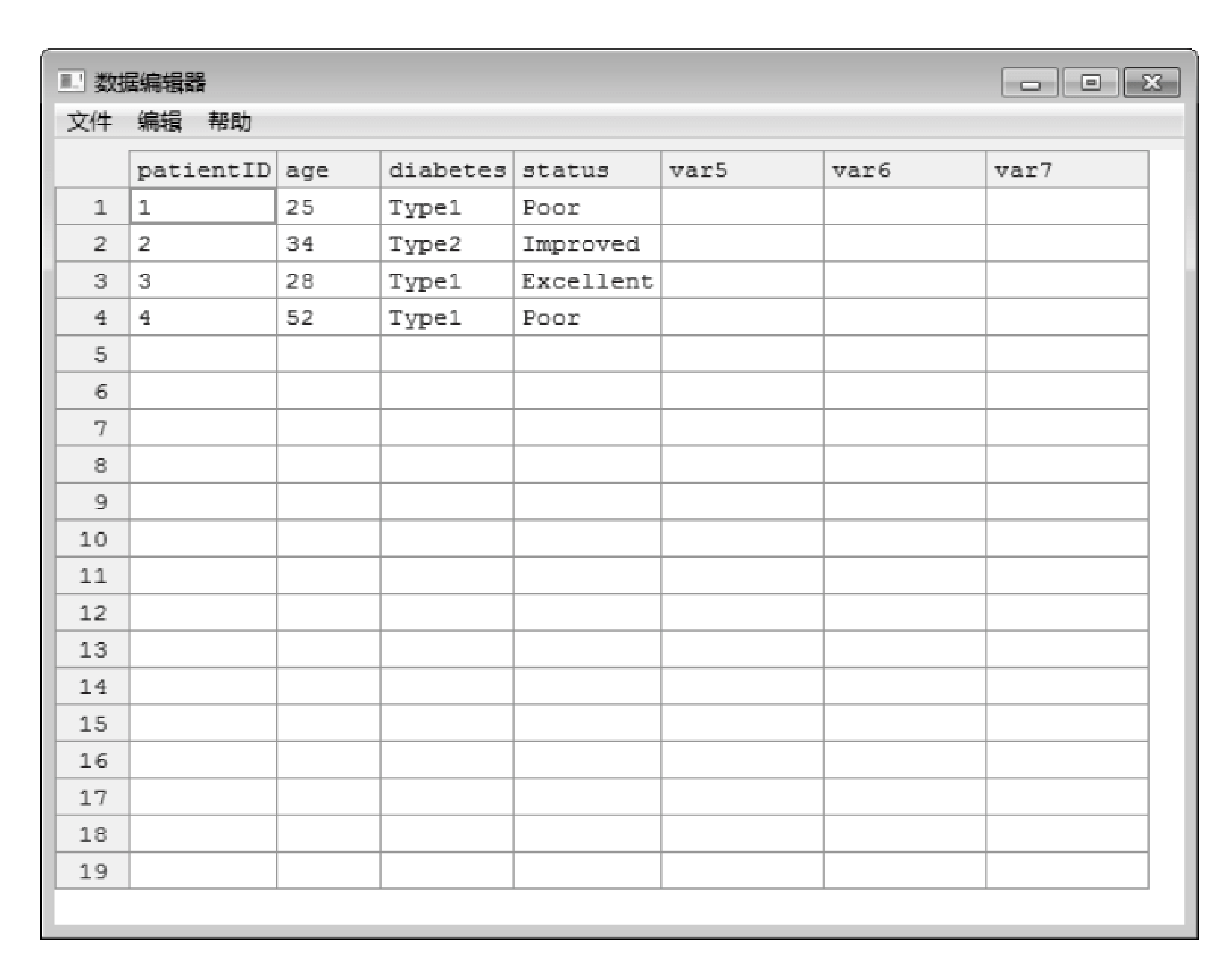


图 4.2　病人登记信息在 R 中的窗口展示

4.2.2　数据的运算

R 语言可以作为一个运行计算并显示结果的“大计算器”。base 包里面包含几乎所有科学计算的函数。表 4.2 列出了部分基础运算函数。

表 4.2　基础运算函数

简单数学运算		常用的函数		逻辑运算	
＋	加法	abs	绝对值	＞	大于
－	减法	sign	符号函数	＞＝	大于等于
*	乘法	log	自然对数	＜	小于
/	除法	exp	指数	＜＝	小于等于
^	乘方	sqrt	平方根	＝＝	等于
％*％	矩阵相乘	sin	正弦函数	&	与
％％(mod)	取余数	cos	余弦函数	|	或
％/％	整除	tan	正切函数	!	非

下面给出了一些简单的运算操作及说明。

1. 简单数学运算

```
> 1 + 1
[1] 2
> 3 % % 2
[1] 1
```

说明：％％是整除运算。

2. 函数运算

```
> sign(3)
[1] 1
> sign( - 3)
[1]  - 1
```

说明：sign 函数是符号运算，若是正数，返回值为 1；若为 0，返回值为 0；若为负数，返回值为－1。

3. 逻辑运算

```
> 3 >= 2
[1] TRUE
> 2 > 3
[1] FALSE
> 3!= 6
[1] TRUE
```

说明：逻辑运算的返回值是 TRUE 或者 FALSE，经常用于函数体里面的 if 语句判断。

4.2.3 数据的导入

对于不同类型的数据，R 提供了多种导入方式，包括键盘直接输入，对文本数据、Excel、SAS 等类型数据的导入及对数据库的访问等。

1. 键盘输入

R 提供了 edit 函数实现通过键盘进行数据的输入，调用该函数会出现一个数据编辑器，在编辑器中可以通过键盘手动输入数据。键盘输入数据分为以下两个步骤：

(1) 建立一个矩阵或数据框，设置变量名及变量模式。

(2) 调用数据编辑器，输入并保存数据。

```
#创建数据框 mydata
> mydata <-  data.frame(a = numeric(0),b = character(0),d = numeric(0))
#使用 edit()调用文本编辑器并输入相应数据,单击变量名可以对变量的类型进行更改,结果如图 4.3 所示
mydata <-  edit(mydata)
#显示数据 mydata
> mydata
    a   b   d
1   1   a   10
2   2   b   20
3   3   c   30
```

可以再次使用 edit()调用数据编辑器对数据进行编辑修改。

2. 文本数据的导入

对于文本数据使用 read.table()导入，有两种操作方法，一种是直接利用导入命令，函

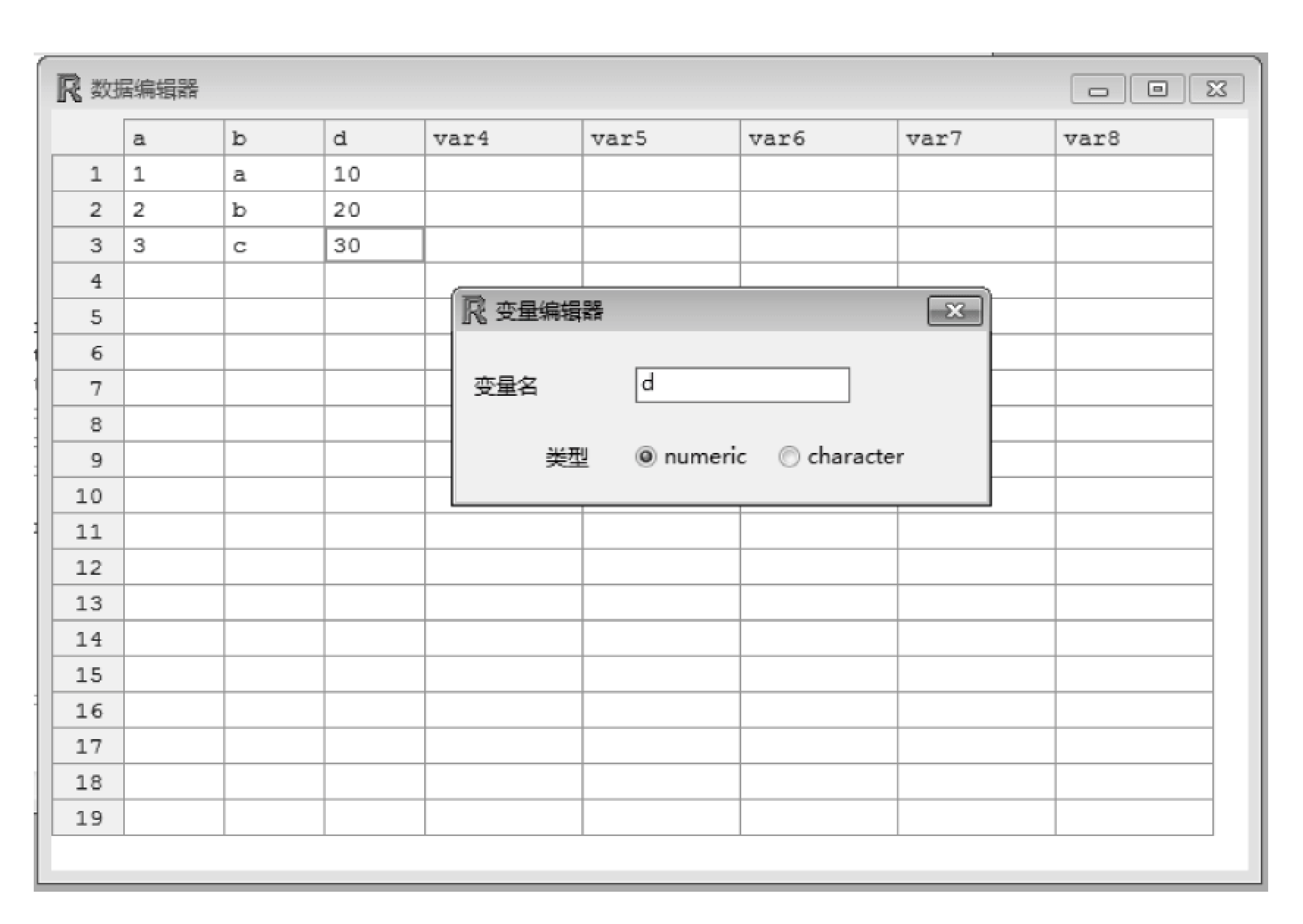


图 4.3　数据编辑器示意图

数语法为：

```
read.table (file, header = T/F, sep = "delimiter", row.names = "name")
```

其中，file 表示要导入的文本文件，该文本文件是带有分隔符的；header 表示是否读取文本文件第一行作为变量名，若为 TRUE，则文本的第一行作为变量名；sep 表示分隔符类型，默认值为 sep＝""，表示分隔符可为一个或多个空格、制表符、换行符或回车符；row.names 表示行标识符的变量，是一个可选参数。例如：

```
> X <- read.table("D:\\Users\\rtest.csv",header = T,sep = "/t",row.names = "r")
```

该命令表示从 Users 文件夹中读取了 rtest 文件保存为 X 数据框，并且读取了第一行作为变量名，该文件是以制表符为分隔符的，行标识符为 r。

另外一种方法是利用剪贴板。首先选中并复制将要导入的数据，然后使用 read.table(“clipboard”)命令进行数据的导入。

3. Excel 数据的导入

R 读取 Excel 数据的方法很多，一种简单的方法是先将 Excel 数据转换成文本文件(.csv)，再按照上节读取文本文件的方法读取。此外，还可以通过 R 语言包进行读取。下面介绍 Excel(.xlsx)文件的读取方式，代码如下：

```
> install.packages("xlsx")     # 安装 xlsx 包
> library(xlsx)     # 加载 xlsx 包
X <- read.xlsx("D:\\Users\\rtest.xlsx",1,header = T)    # 调用函数 read.xlsx()实现 Excel
数据的导入
```

至此,完成了 Excel 数据的读取。其中 1 表示读取 Excel 表中的第一个 Sheet。

4. 其他类型数据的导入

对于 SPSS、SAS 及 Stata 数据的导入,R 提供了相应的包。例如,foreign 包中的函数 read. spss()、read. ssd()及 read. dta()可以分别读取 SPSS 数据、SAS 数据和 Stata 数据;还可以分别调用 Hmisc 包中的函数 spss. get()和 sas. get()对 SPSS 及 SAS 类型的数据进行读取。在调用函数前需安装和加载相应的包。而对于 SAS 数据,可以先将其另存为文本文件(以逗号为分隔符),然后使用文本文件的读取方法进行读取。

例如,应用 foreign 包中的函数 read. spss()读取 SPSS 文件,代码如下:

```
> install.packages("foreign")          # 安装 foreign 包
> library(foreign)                     # 加载 foreign 包
> read.spss('D:\\Users\\居民储蓄调查数据.sav')          # 调用 read.spss()读取 spss 文件
```

5. 数据库的访问

R 提供了很多关系型数据库管理系统的接口,比如 MySQL、Oracle、Microsoft SQL Server 等。R 访问数据库,克服了 R 对大数据存储的限制,充分发挥了 R 的数据分析功能,大大提高了 R 对大数据分析的性能。

在 R 中通过 RODBC 包进行数据库的访问。该方法使得 R 能够访问任意具有 ODBC 驱动的数据库。首先需要安装 RODBC 包,表 4.3 列出了该包中一些基本的函数。

表 4.3 RODBC 包中的函数

函　　数	函数描述
odbcConnect(dsn,uid=" ",pwd=" ")	建立与 ODBC 数据库的连接
sqlFetch(channel,sqltable)	将 ODBC 数据库中的指定表读取到数据框中
sqlQuery(channel,query)	向 ODBC 数据库提交查询并且返回相应结果
sqlSave(channel, mydf, tablename = sqtable, append=FALSE)	将数据框写入 ODBC 数据库相应的表中,若 append=TRUE 则为数据的更新
sqlDrop(channel,sqtable)	将 ODBC 数据库中指定表删除
close(channel)	将连接关闭

RODBC 包提供 R 读取、编辑数据库中数据的功能,实现 R 与数据库间的相互通信。

4.3 数据的管理

R 有类似于 SQL 中的增删改查(insert、delete、update、select)等对数据库操作,下面进行具体说明。

4.3.1 数据排序

有些情况下,查看排序后的数据集可以获得相当多有用的信息。在 R 中可以使用 order 函数对数据框进行排序,默认为升序。在排序变量的前边加一个减号即可得到降序的排序结果。下面将利用上一节输入的数据框 patientdata 进行举例。

```
#创建一个新的数据集,其中各行按照病人的年龄升序排序
> newdata <- patientdata[order(patientdata$age),]
> newdata
  patientID age diabetes status
1         1  25    Type1      Poor
3         3  28    Type1 Excellent
2         2  34    Type2  Improved
4         4  52    Type1      Poor
#各行先按照病人的病情升序排序,同等病情的再按照年龄升序排序
> attach(patientdata)
> newdata2 <- patientdata[order(status, age),]
> newdata2
  patientID age diabetes status
3         3  28    Type1 Excellent
2         2  34    Type2  Improved
1         1  25    Type1      Poor
4         4  52    Type1      Poor
> detach(patientdata)
```

4.3.2 数据集的合并

数据集的合并(Insert)在实际运用中十分常见,比如进行问卷调查,后期补录了一组变量,需要与前期的数据进行列合并;几个调研员调查的样本要进行汇总合并等。本节将分别展示向数据框中添加列(变量)和行(样本)的方法。

1. 添加列

要横向合并两个数据框(数据集)使用 merge 函数。在多数情况下,两个数据框是通过一个或多个共有变量进行联结的(即一种内联结,Innerjoin)。例如:

```
total <- merge(dataframeA, dataframeB, by = "ID")  #将 dataframeA 和 dataframeB 按照 ID 进行合并
```

类似地:

```
total <- merge(dataframeA, dataframeB, by = c("ID","Country"))  #将两个数据框按照 ID 和 Country 进行合并
```

类似地,横向联结通常用于向数据框中添加变量。若要直接横向合并两个矩阵或数据框,并且不需要指定一个公共索引,则可以直接使用 cbind 函数。

```
total <- cbind(A, B)
```

这个函数横向合并对象 A 和对象 B。需要注意的是,使用 cbind 函数合并对象,其中每个对象必须拥有相同的行数,且要以相同顺序排序。

2. 添加行

要纵向合并两个数据框(数据集)使用 rbind 函数。

```
total <- rbind(dataframeA, dataframeB)
```

两个数据框必须拥有相同的变量，但它们的顺序不必一定相同。如果 dataframeA 中拥有 dataframeB 中没有的变量，在合并它们之前需做以下某种处理：

（1）删除 dataframeA 中的多余变量。

（2）在 dataframeB 中创建追加的变量并将其值设为 NA（缺失）。

纵向联结通常用于向数据框中添加样本。

4.3.3 剔除变量

剔除变量（Delete）的原因有很多，比如某个变量中有若干缺失值，在进行分析之前就需要将其删除。下面是一些剔除变量的方法，使用之前创建的数据集进行举例。

```
> newdata3 <- patientdata[, -2]
> newdata3
  patientID diabetes    status
1         1    Type1      Poor
2         2    Type2  Improved
3         3    Type1 Excellent
4         4    Type1      Poor
```

上面的例子剔除了数据集 patientdata 中的变量 age。其中，方框[,-2]表示提取除去第二列的所有行。同样，可以用下面的一种方法剔除变量 age。

```
> newdata4 <- patientdata
> newdata4 $ age <- NULL
> newdata4
  patientID diabetes    status
1         1    Type1      Poor
2         2    Type2  Improved
3         3    Type1 Excellent
4         4    Type1      Poor
```

语句的含义是将 age 列设为未定义（NULL）。需要注意的是，NULL 与 NA（表示缺失）是不同的。

丢弃变量是保留变量的逆向操作，选择哪一种方式进行变量筛选依赖于两种方式的编码难易程度。比如，如果需要剔除绝大多数的变量，那么选择保留需要留下的变量，操作起来会更简单，反之亦然。

4.3.4 数据集提取

R 拥有强大的索引特性，可以用于访问对象中的元素，也可利用这些特性对变量或观测进行选入和排除。

1. 提取变量（列）

从一个大数据集中选择有限数量的变量（列）来创建一个新的数据集，在实践应用中广

泛存在。在前面的小节中提到,数据框中的元素是通过以下代码提取的,提取变量的方法就是选择被提取变量的列号,方法同前文,此处不再赘述。

```
dataframe(row indices, column indices)
```

2. 提取样本(行)

提取或剔除样本(行)通常是数据准备和数据分析的一个关键方面。比如,在进行统计调查时通常需要甄别变量,进行对比分析;在一些设置了实验组和对照组实验研究中,需要提取一类样本进行分析等。和提取变量一样,可以通过选择行号来提取样本,也可以选择满足所需条件的样本。例如:

```
> newdata5 <- patientdata[which(patientdata$status == "Poor" & patientdata$age > 30), ]
> newdata5
  patientID age diabetes status
4         4  52    Type1   Poor
> attach(patientdata)
> newdata6 <- patientdata[which(status == "Poor" & age > 30), ]
> detach(patientdata)
> newdata6
  patientID age diabetes status
4         4  52    Type1   Poor
```

在上述示例中提取了所有 30 岁以上并且状态为 Poor 的病人样本。

如果使用了 attach 函数,则可以直接通过变量名称进行调用,而不需要在变量名前加上数据框名称。调用数据集结束后,使用 detach 函数进行释放。

4.3.5 subset 函数

前面几节中的示例描述了逻辑型向量和比较运算符在 R 中的解释方式,理解这些例子的工作原理将有助于对 R 代码的解读。而使用 subset 函数可以使选择变量的操作变得更为简单,例如:

```
> newdata7 <- subset(patientdata, age >= 30, select = 2:4)
> newdata7
  age diabetes   status
2  34    Type2 Improved
4  52    Type1     Poor
> newdata8 <- subset(patientdata, status == "Poor",select = 2:4)
> newdata8
  age diabetes status
1  25    Type1   Poor
4  52    Type1   Poor
```

在第一个示例中选择了所有 age 值大于等于 30 的行,保留了变量 2~4 列;在第二个示例中选择了所有状态为 Poor 的病人,并保留了变量 2~4 列。

4.4 常用函数

在使用 R 软件进行数据分析处理时，会经常使用一些函数命令来查看数据的特征和基本结构，这将有助于避免数据处理中的错误。一些常用的函数如表 4.4 所示。

表 4.4 常用函数

函　　数	功　　能
length()	显示对象元素的个数
dim()	显示对象的维度
str()	显示对象的结构
class()	显示对象的类
mode()	显示对象的模式
names()	显示对象各成分的名称
c(对象 a,对象 b)	连接 a、b 两个对象
cbind(对象 a,对象 b)	按列合并 a、b 两个对象
rbind(对象 a,对象 b)	按行合并 a、b 两个对象
head()	列出对象前 6 个样本
tail()	列出对象后 6 个样本
ls()	显示当前对象列表
rm()	删除对象
fix()	编辑对象

mode()和 length()是类型和长度属性函数，使用率比较高。

```
> mode(c(1,3,5))
[1] "numeric"
> mode(c(1,3,5)> 5)
[1] "logical"
> a <- 1:10
> length(a)
[1] 10
> length(a) <- 5#缩短长度,将得到子集
> a
[1] 1 2 3 4 5
> A <- matrix(seq(1:12),ncol = 3)
> dim(A)
[1] 4 3
> A
     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12
```

对矩阵、数据框、数组的长度查询需要使用 dim()，其返回值的第一个元素代表行数，第二个元素代表列数。

CHAPTER 5

第5章

R语言绘图

R 提供了丰富的可视化函数，绘图功能强大。本章介绍 R 的一些基本绘图技术，首先介绍基本的绘图参数，包括符号、颜色、标题、尺寸及图形的组合等；然后描述常用的绘图函数，如条形图、直方图、核密度图等。

5.1 绘图参数

绘图参数(Graphical Parameters)提供了丰富的绘图选项。R 的常用绘图参数包括字体、颜色、坐标轴、标题等，可以通过以下两种方式进行设定和修改。

(1) 在高级绘图函数(如 hist/boxplot/plot 等)中直接指定，进行临时性参数设置，例如：

```
> hist(mtcars $ mpg, col.lab = "red")
```

(2) 通过 par 函数进行全局性参数设置，可以通过修改图形参数的选项自定义图形的某些特征。该方式设定的参数值，如果不再修改，在结束会话前都是有效的。其调用命令格式为：

```
par(optionname = value, optionname = name, … )
```

```
> par()               # 查看当前绘图参数设置
> opar <- par()       # 保存当前设置
> par(col.lab = "red") # 设置坐标轴标签为红色
> hist(mtcars $ mpg)  # 利用新的参数绘图
> par(opar)           # 恢复绘图参数的原始设置
```

下面通过一个具体的例子进行演示。表 5.1 给出了一个假设出来的数据集，描述的是空调价格对需求的影响情况。

表 5.1 某品牌空调价格需求表

价格 p(千元)	需求量 q(万台)
1	70
2	69
3	63
4	60
5	58

先将数据导入 R 中，代码如下：

```
> p<-1:5
> q<-c(70,69,63,60,58)
```

现使用以下代码创建一幅描述该品牌空调价格及其需求量的响应关系图，如图 5.1 所示。

```
> plot(p,q,"b")
```

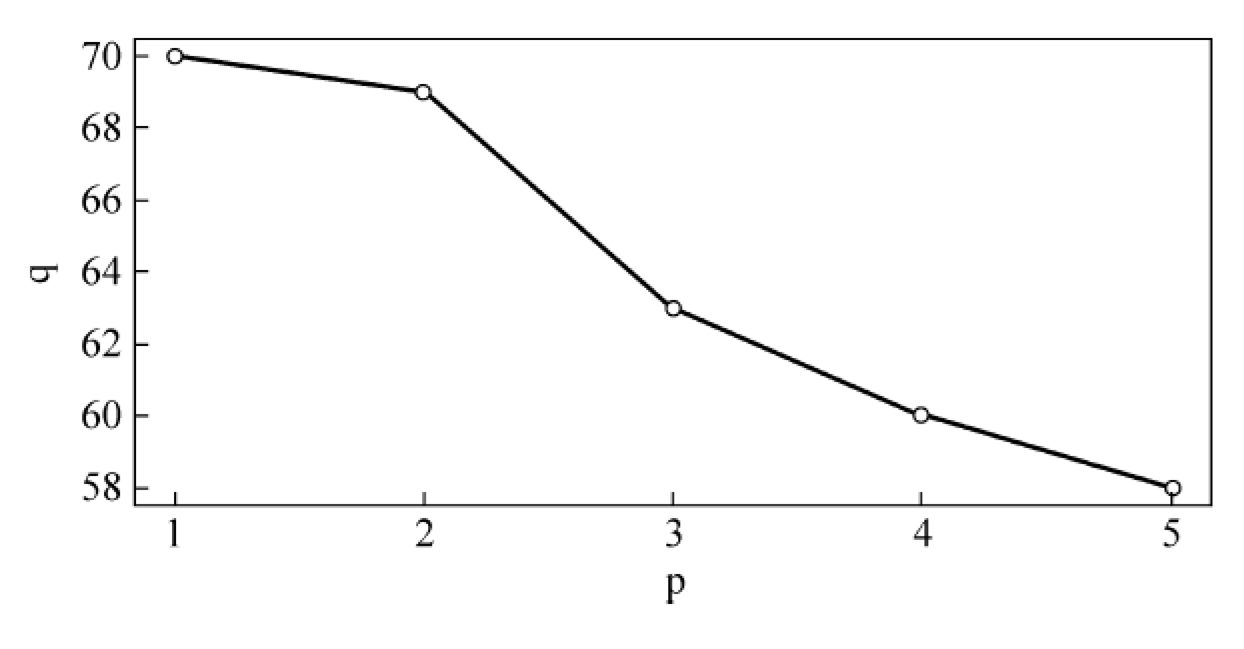


图 5.1 空调价格及其需求量响应关系图

通过使用以下代码对参数进行修改，可以得到图 5.2 所示的效果。

```
> plot(p,q,"b",lty=3,pch=17)
```

plot 函数的参数表示将线条类型修改为虚线（lty＝3），并将点符号改为实心三角（pch＝17）。也可以使用 par()进行永久修改。

```
> opar<-par()                  #保存当前设置
> par(lty=3,pch=17)            #设定默认的线条为虚线,符号为实心三角形
> plot(p,q,"b")
> par(opar)                    #恢复绘图参数的原始设置
```

为了更详尽地展现 R 语言的强大绘图能力，下面将分章节介绍其绘图参数。

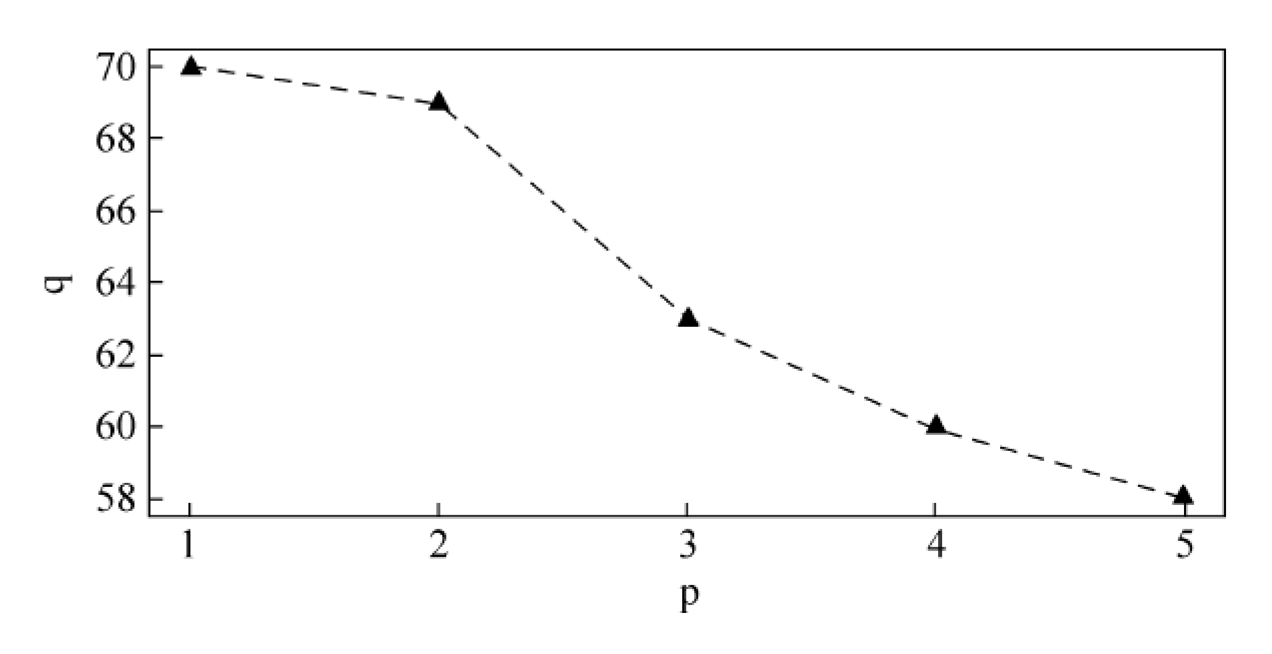
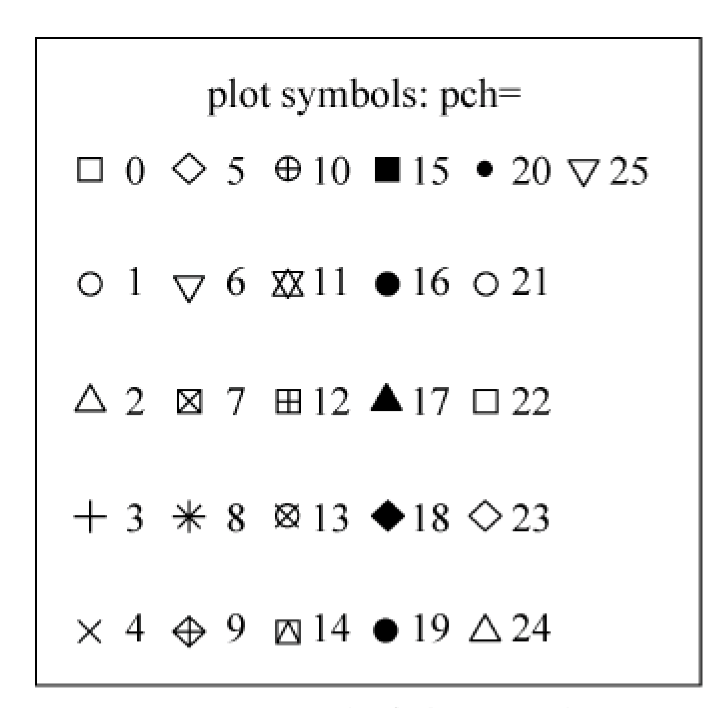


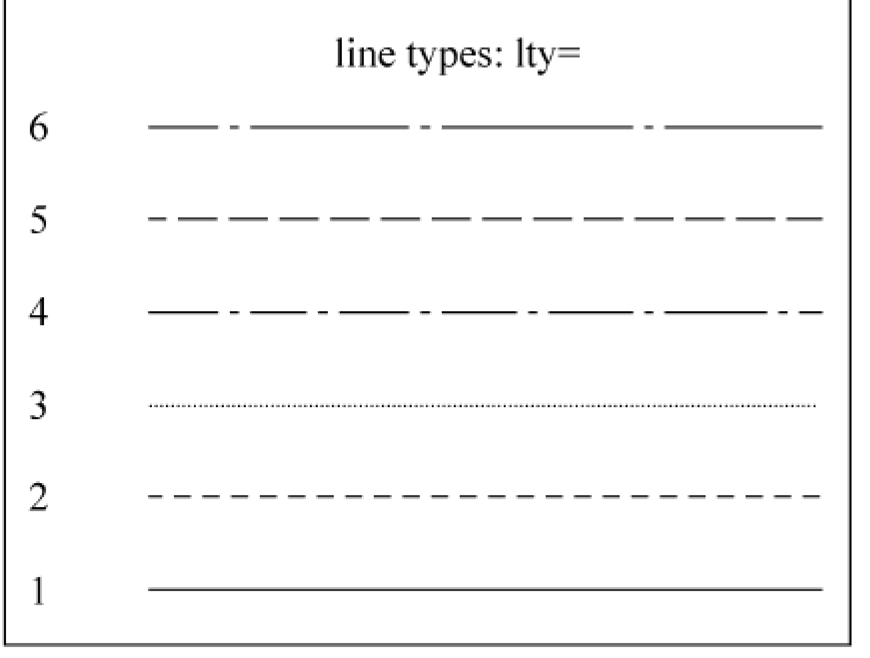
图 5.2　空调价格及其需求量关系添加效果图

5.1.1　符号、线条与颜色

R 语言提供了丰富的绘图参数，可以使用这些绘图参数设置符号、线条及颜色类型。常用的绘图参数如图 5.3 和表 5.2 所示[①]。



(a) plot函数点的参数设置图



(b) plot函数线的参数设置图

图 5.3　plot 函数点和线的参数设置图

表 5.2　点符号、线条与颜色参数表

参　　数	功　　能
pch	指定绘制点时使用的符号，R 中共有 26 种点符号供选择
lty	指定线条类型，R 中共有 7 种线型供选择。0＝blank（空白）、1＝solid（实线）（default）、2＝dashed（虚线）、3＝dotted（点线）、4＝dotdash（点和虚线）、5＝longdash（长虚线）、6＝twodash（长短虚线）
lwd	指定线条宽度。lwd 是以默认值的相对大小来表示的（默认值为 1）。例如，lwd＝2 将生成一条两倍于默认宽度的线条
cex	指定符号的大小（默认值为 1）。例如，cex＝2 将生成两倍于默认值的符号。 cex. axis：坐标轴刻度标记的缩放比例。 cex. lab：坐标轴标题的缩放比例。 cex. main：指定主标题的缩放比例。 cex. sub：子标题的缩放比例

① Robert I. Kabacoff. R 语言实战[M]. 高涛，肖楠，陈钢，译. 北京：人民邮电出版社，2013.

续表

参　数	功　能
font	用于指定绘图使用的字体样式。1＝常规，2＝粗体，3＝斜体，4＝粗斜体。 font. axis：坐标轴刻度文字的字体样式。 font. lab：坐标轴标签（名称）的字体样式。 font. main：标题的字体样式。 font. sub：副标题的字体样式。 ps：字体磅值（1 磅约为 1/72 英寸）。文本的最终大小为 ps * cex。 family：绘制文本时使用的字体族。标准的取值为 serif（衬线）、sans（无衬线）和 mono（等宽）
col	指定绘图颜色。 col. axis：坐标轴刻度标记的颜色。 col. lab：坐标轴标题的颜色。 col. main：主标题的颜色。 col. sub：子标题的颜色。 fg：设置前景色。 bg：设置背景色。 在 R 中可以通过颜色下标、颜色名称、十六进制的颜色值、RGB 值或 HSV 值来指定颜色。举例来说，col＝1、col＝"white"、col＝"＃FFFFFF"、col ＝ rgb（ 1,1,1 ）和 col ＝ hsv（ 0,0,1 ）都表示白色的等价方式。函数 rgb 可基于红—绿—蓝三色值生成颜色，而函数 hsv 则基于色相—饱和度—亮度值来生成颜色。
pin	以英寸表示的图形尺寸（宽和高）
mai	边界大小，顺序为“下、左、上、右”，单位为英寸
mar	边界大小，顺序为“下、左、上、右”，单位为英分。默认值为 c(5，4，4，2) ＋ 0.1

下面将通过具体的例子展示参数的设置对图形的影响效果，具体如下：

```
> u<-1:25
> plot(u,pch = u,col = u,cex = 2)
```

图 5.4 展示了 R 语言的 25 种点符号及 8 种基本色。plot 绘制的图像，x 轴是序号，y 轴是 u 的值，pch＝u 表示点符号设定为 1～25 号，col＝u 表示颜色设定为 1～25，因为 R 中用数字代表的颜色有 8 种，所以自动进行循环。

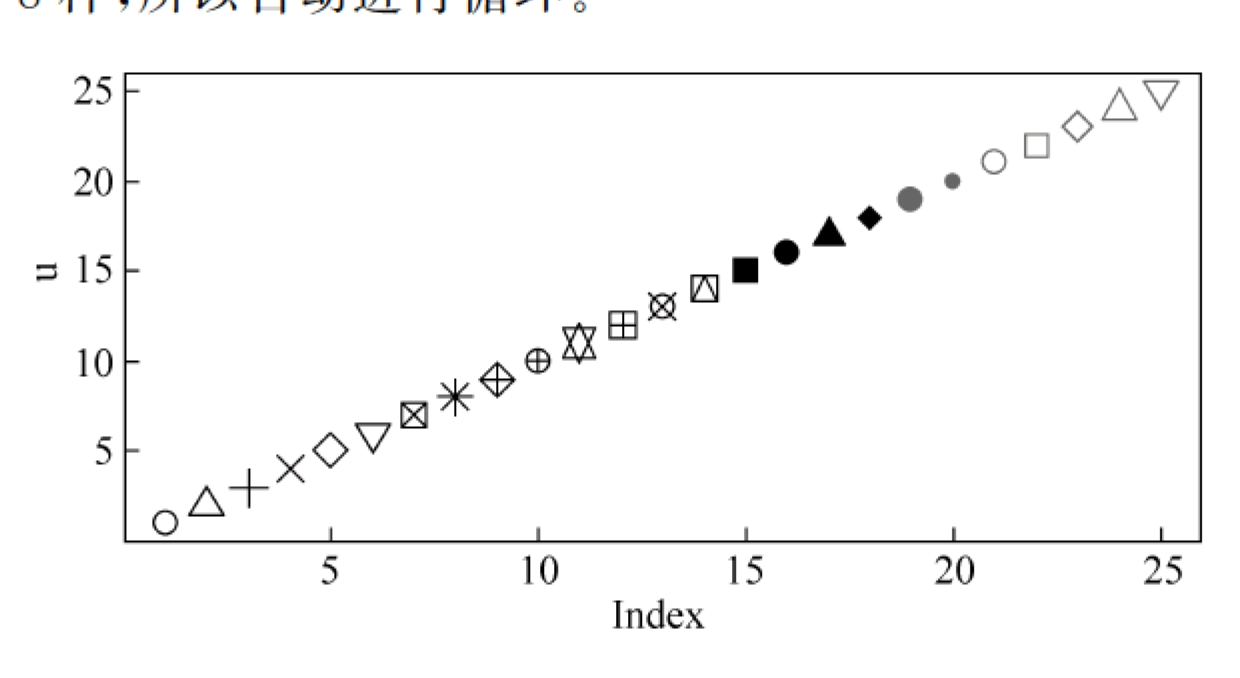


图 5.4　plot 点参数变化图

```
> matplot(matrix(1:60,10,6),lty = 1:6,lwd = 2,type = 'l')
```

图 5.5 展示了 R 语言的 6 种线条类型。matplot 绘制矩阵的图像，matrix(1:60,10,6)表示一个 10 行 6 列元素为 1～60 的矩阵，x 轴是矩阵行号，y 轴是元素的值，每条线是每一列的元素勾勒出来的。lty=1:6 表示线条类型设定为 1～6 号，lwd=2 表示线条宽度设定为默认值的 2 倍，type='l'表示画的为实线。

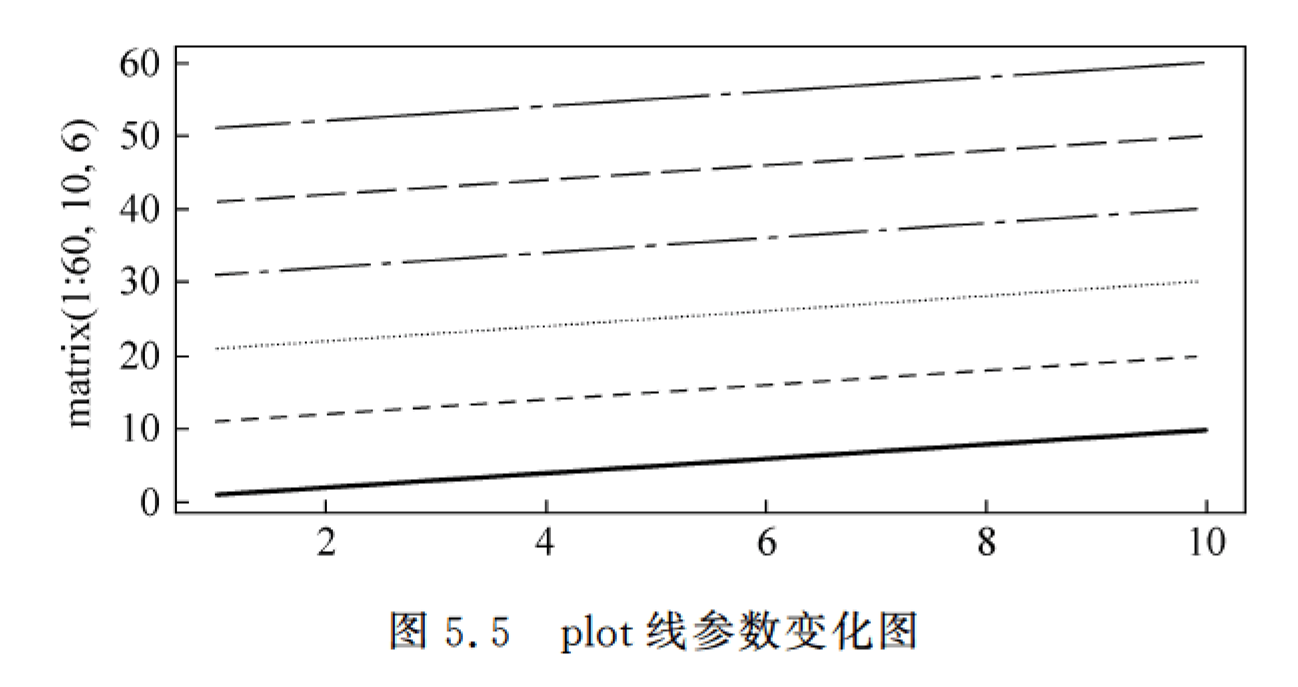


图 5.5　plot 线参数变化图

关于颜色的设定，colors()可以返回所有可用颜色的名称。R 中也有多种用于创建连续型颜色向量的函数，包括 rainbow()、heat. colors() 、terrain. colors() 、topo. colors()及 cm. colors()，具体请参考这些函数的帮助文件。

例如，rainbow(10)可以生成 10 种连续的“彩虹型”颜色。多阶灰度色可使用 gray()生成，这时要通过一个元素值为 0 和 1 之间的向量来指定各颜色的灰度。gray(0:10/10)将生成 10 阶灰度色。颜色参数设置对比见图 5.6。

```
> n <- 10
> mycolors <- rainbow(n)            # 把 n 种彩虹色赋值给 mycolors
> pie(rep(1,n),col = mycolors)      # 画 n 块分区的饼图，颜色设定为 mycolors
> mygrays <- gray(0:n/n)            # 把 n 种灰度赋值给 mygrays
> pie(rep(1,n),col =  mygrays)      # 画 n 块分区的饼图，颜色设定为 mygrays
```

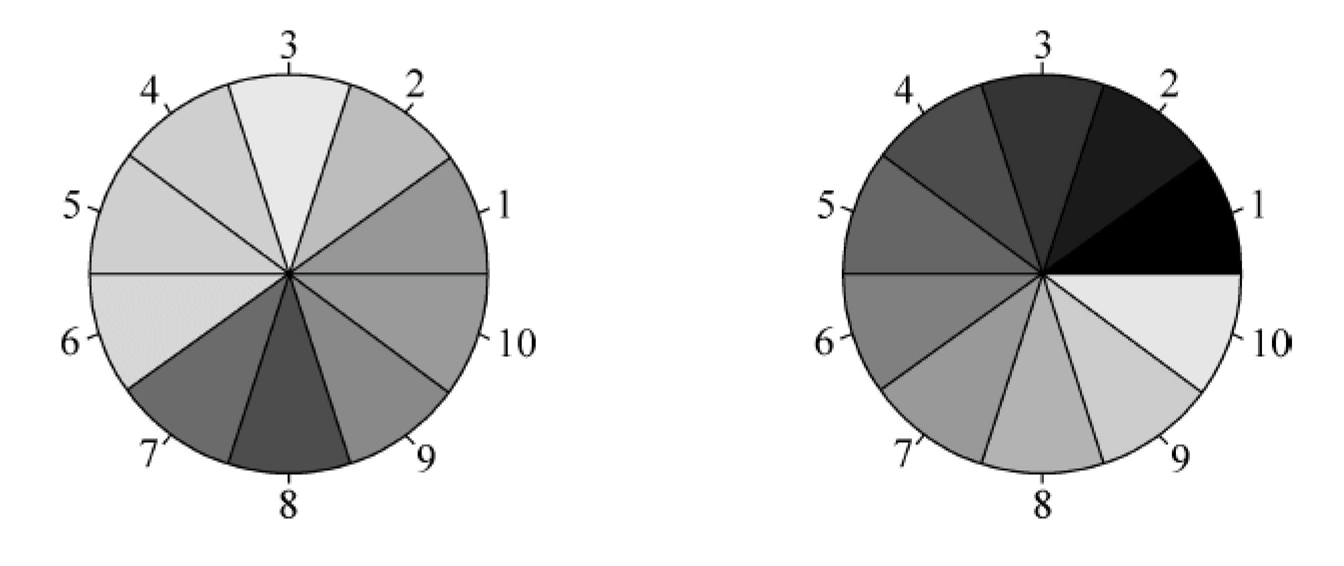


图 5.6　颜色参数设置对比

5.1.2　标题、坐标轴与图例

在图形上可以添加标题(main)、副标题(sub)、坐标轴标签(xlab、ylab)并指定坐标轴范围(xlim、ylim)。表 5.3 列出了一些常用的标签说明。

表 5.3 标签说明表

参　　数	功　　能
标题 title(main=,sub=,xlab=,ylab=)	
main	主标题
sub	副标题
xlab	x 轴标签
ylab	y 轴标签
坐标轴 axis(side,at=,labels=,lty=,col=,las=,tck=,…)	
side	坐标轴绘制位置(1：下；2：左；3：上；4：右)
at	需要绘制刻度线的位置(x/y)
labels	刻度线旁边的文字标签
lty	坐标轴线条类型
col	线条和刻度线的颜色
las	0：标签平行于坐标轴；2：标签垂直于坐标轴
tck	刻度线的长度(默认值为−0.01)
图例 legend(location,title,legend)	
location	有许多方式可以指定图例的位置，可以直接给定图例左上角的 x、y 坐标，也可以执行 locator(1)，然后通过鼠标单击给出图例的位置，还可以使用关键字 bottom、bottomleft、left、topleft、top、topright、right、bottomright 或 center 放置图例
title	图例标题的字符串(可选)
legend	图例标签组成的字符型向量

继续之前空调价格对需求影响情况的例子，使用高级绘图函数直接加入参数，代码和图形结果如下：

```
> p<-1:5
> q<-c(70,69,63,60,58)
> plot(p,q,"b",col = 'blue',
       lty = 2,pch = 2,lwd = 2,
       main = "某品牌空调价格需求曲线",
       sub = "假设的数据",
       xlab = "价格",ylab = "需求量",
       xlim = c(1,5),ylim = c(58,70)
       )
```

图 5.7 加上了标题、坐标标签，使得图片所表示的含义和代表的变量一目了然。

也可以通过 title()、axis()、legend()完成以上参数设定，但是某些高级绘图函数已经包含了默认的设定，需要通过以下办法先进行消除。其中，针对标题和标签，可以通过在 plot()语句或单独的 par()语句中添加 ann=FALSE 来移除它们；对于自动生成的坐标轴，axes=FALSE 将禁用全部坐标轴，参数 xaxt="n"和 yaxt="n"将分别禁用 x 轴或 y 轴(会留下框架线，只是去除了刻度)，利用之前的例子进行说明。图 5.8 是修改参数后的空调价格需求曲线图。

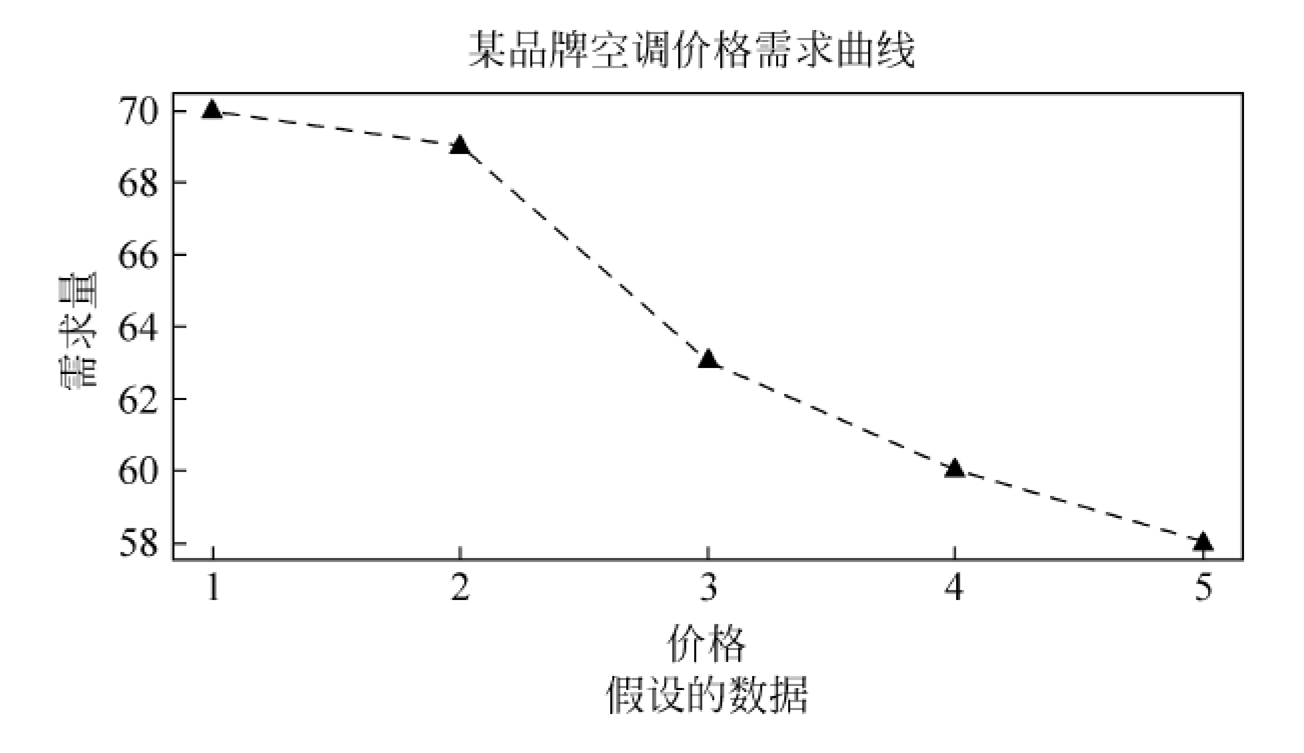


图 5.7　空调价格需求曲线

```
> p<-1:5
> q<-c(70,69,63,60,58)
> plot(p,q,"b",col = 'blue',lty = 2,pch = 2,lwd = 2,ann = F,axes = F)
> title(main = "某品牌空调价格需求曲线",col.main = 'red',
+        sub = "假设的数据",col.sub = 'blue',
+        xlab = "价格",ylab = "需求量",
+        )
> axis(1,at = p,labels = p,col.axis = 'red',las = 1)
> axis(2,at = q,labels = q,col.axis = 'red',las = 2)
> legend("topright",inset = 0.05,title = "图例",
"价格需求线",col = 'blue',lty = 2,pch = 2,lwd = 2
)
```

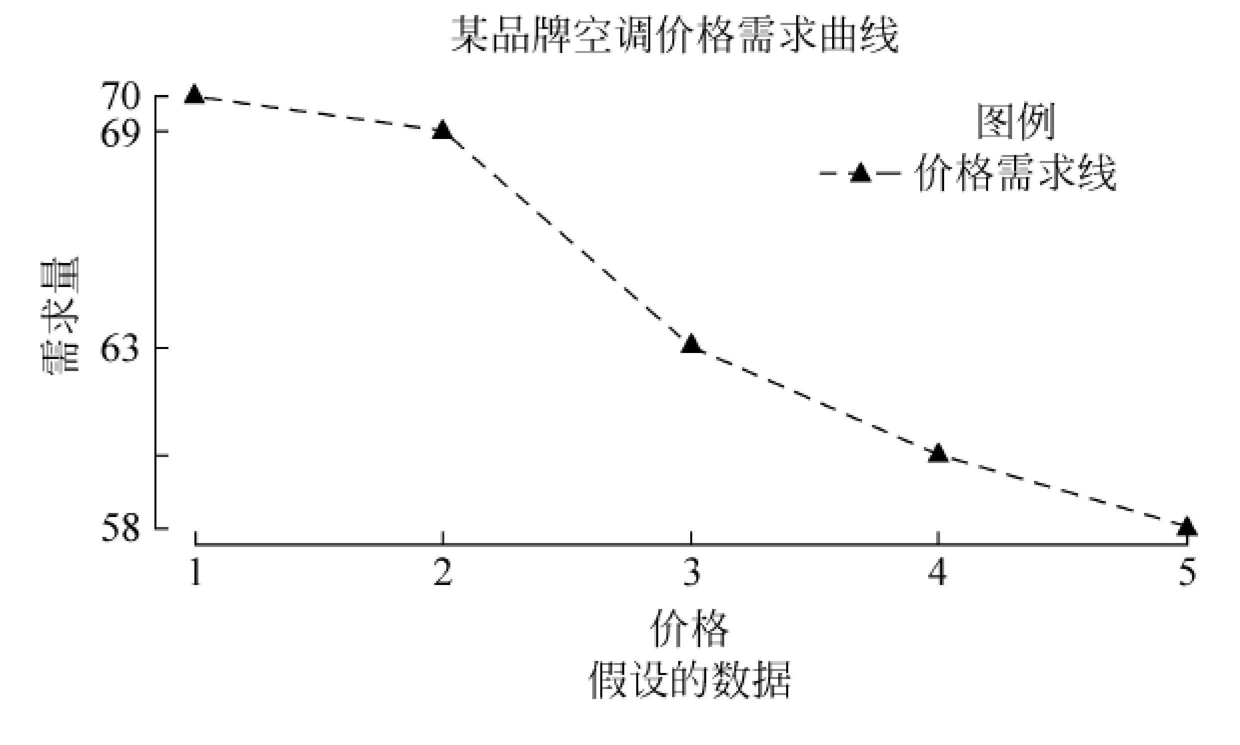


图 5.8　修改参数后的空调价格需求曲线图

5.1.3　文本属性

运用图形参数还可以设置字体、字号等的类型。表 5.4 和表 5.5 列出了一些常用的设置文本属性的参数。

表 5.4 设置文本大小参数表

参　　数	参数描述
cex	表示相对于默认大小缩放的倍数。默认大小的值为 1；值 1.5 为默认值 1 的 1.5 倍，即放大为默认值的 1.5 倍；而值 0.5 则表示缩小为默认值 1 的 0.5 倍，其他数值类似
cex. axis	类似 cex，表示坐标轴刻度文字缩放的倍数
cex. lab	坐标轴标签缩放的倍数
cex. main	标题缩放的倍数
cex. sub	副标题缩放的倍数

表 5.5 设置字体、字号及字样的参数表

参　　数	参数描述
font	设置图形的字体样式，为一整数值。1 表示常规；2 表示粗体；3 表示斜体；4 表示粗斜体；5 表示符号字体
font. axis	坐标轴刻度文字的字体样式
font. lab	坐标轴标签的字体样式
font. main	标题的字体样式
font. sub	副标题的字体样式
ps	字体的磅值
family	对文本进行绘制时的字体族，标准的取值有三种：serif（衬底）、sans（无衬底）、mono（等宽）

例如，如果在绘制图 5.7 所示的空调价格需求曲线图的 R 代码前加上如下语句：

```
> par(font.lab = 3,cex.lab = 0.5,font.main = 4,cex.main = 1.5)
```

那么所绘制图形的坐标轴标签的大小为默认文本大小的 0.5 倍，字体为斜体；标题的大小为默认文本大小的 1.5 倍，字体为粗斜体。效果如图 5.9 所示。

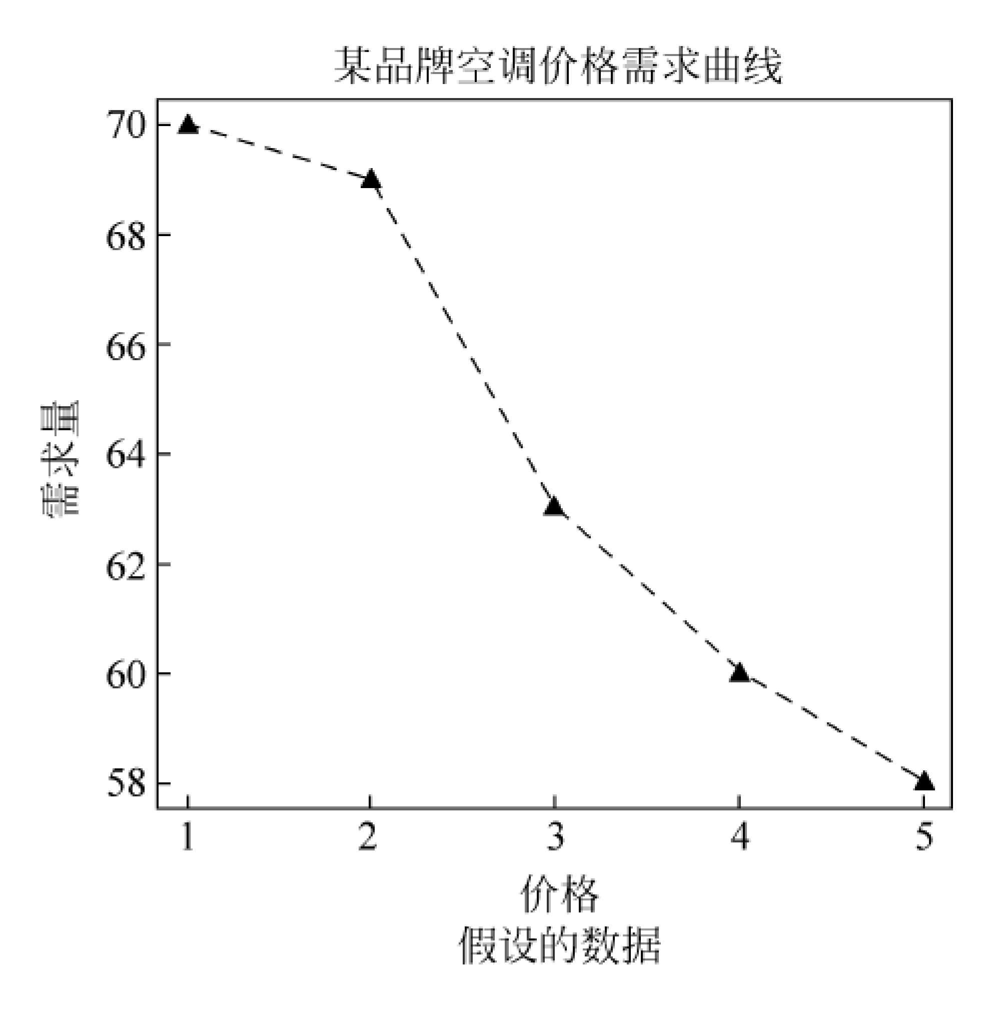


图 5.9 空调价格需求曲线图

5.1.4　图形的组合

在 R 中可以使用 par()或 layout()将多幅图形组合为一幅总括图形，下面分别举例说明。

1. par()

在 par()中，图形参数 mfrow=c(a,b)可以创建按行填充的、行数为 a、列数为 b 的图形矩阵；而图形参数 nfcol=c(a,b)生成的是按列填充矩阵。利用数据集 mtcars 进行绘图，将创建 4 幅图形并将其排布在两行两列中，如图 5.10 所示的图形演示。

```
> attach(mtcars)                # 加载 mtcars 数据集
> opar <- par()                 # 保存当前设置
> par(mfrow = c(2,2))           # 设置两行两列的画纸
> plot(hp,mpg,main = '马力能耗散点图')
> plot(wt,mpg,main = '重量能耗散点图')
> hist(cyl,main = '气缸频数直方图')
> boxplot(wt,main = '重量箱线图')
> par(opar)                     # 恢复原来设置
> detach(mtcars)                # 释放数据集
```

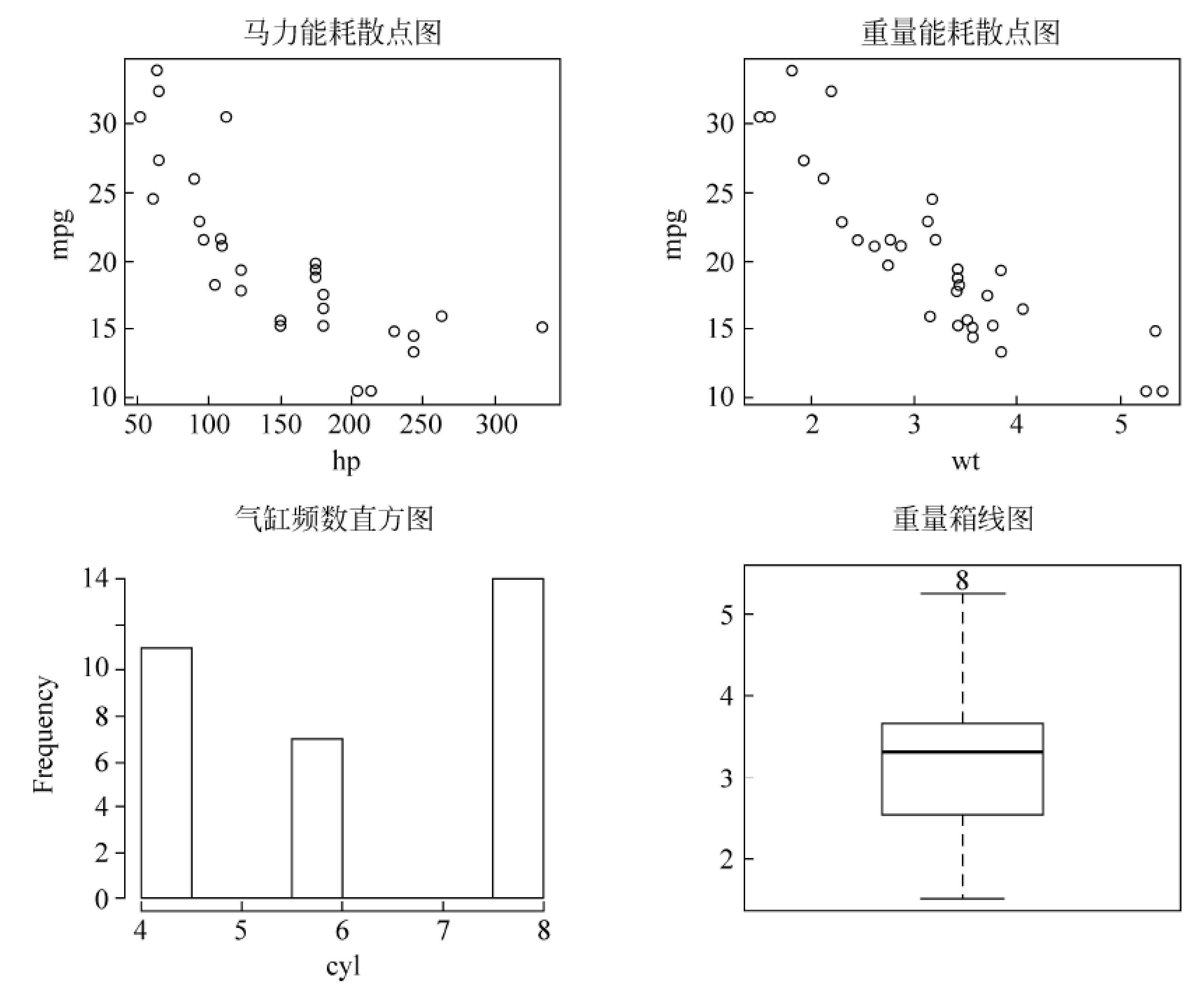


图 5.10　各种常用图形演示

2. layout()

layout()的调用形式为 layout(matrix)，其中的 matrix 为一个矩阵，该矩阵指定了多个要进行组合的图形所在的位置。图 5.11 是 layout 布局演示图。下面用数据集 mtcars 进行举例说明。

```
> attach(mtcars)          # 加载 mtcars 数据集
 # 设置画纸第一行两幅图,第二行放置第三幅图(图 5.11)
> layout(matrix(c(1,2,3,3),2,2,byrow = T))
> plot(hp,mpg,main = '马力能耗散点图')
> plot(wt,mpg,main = '重量能耗散点图')
> hist(cyl,main = '气缸频数直方图')
> detach(mtcars)          # 释放数据集
```

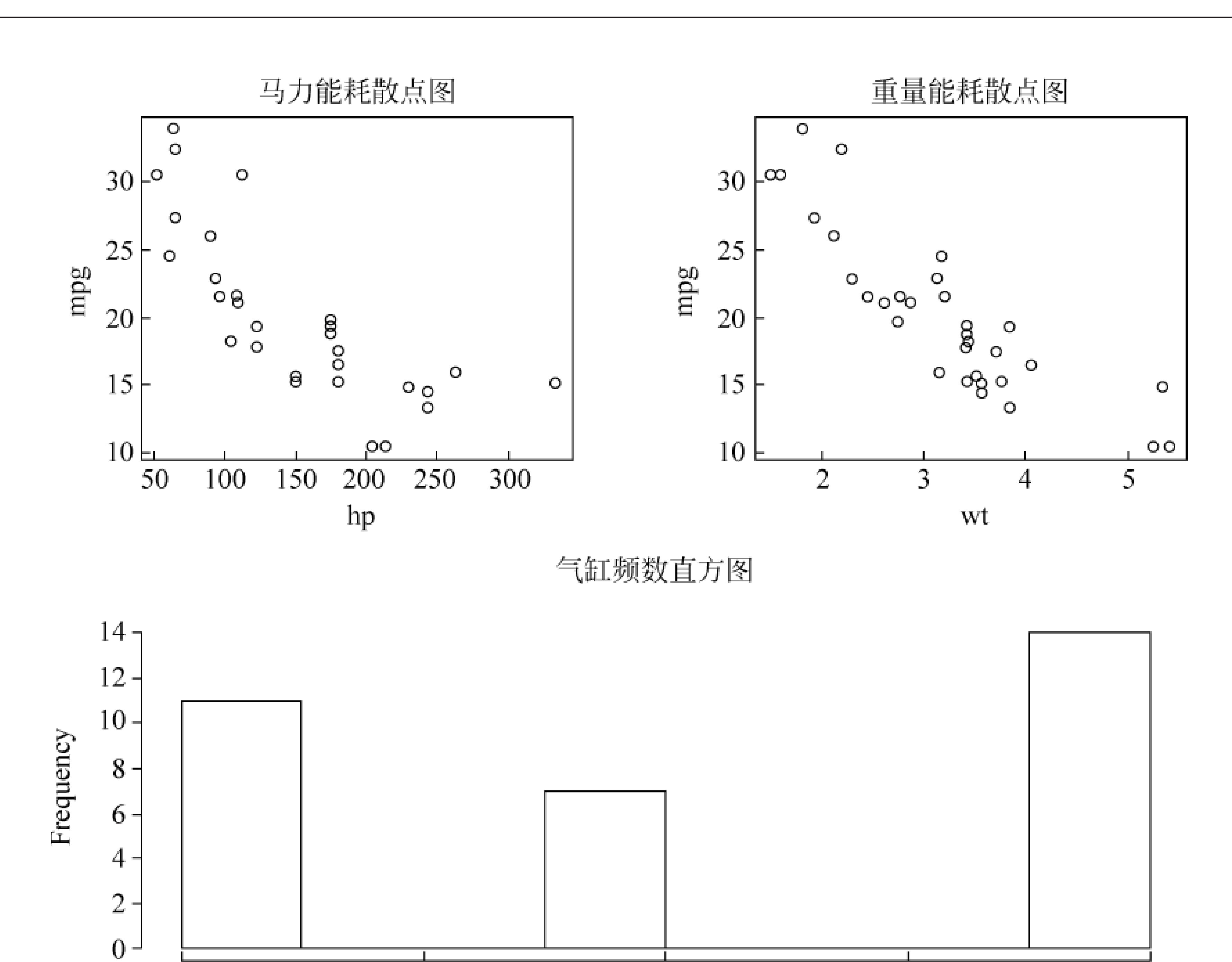


图 5.11 layout 布局演示图

layout(matrix(c(1,2,3,3),2,2,byrow=T))将画纸设定为两行两列、按行排列的矩阵,第一行两个位置分别放置第一幅图和第二幅图,第二行两个位置放置第三幅图。

5.2 高级绘图函数

通过高级绘图函数可以创建不同类型的图形。表 5.6 列出了 R 语言中一些常用的高级绘图函数,本章将对这些常用的高级绘图函数进行举例说明。

表 5.6 高级绘图函数

函 数	功 能	函 数	功 能
plot()	通用二维图	pie()	饼图
boxplot	箱线图	barplot()	条形图
hist()	直方图	dotchart()	散点图

5.2.1　通用二维图

plot( )是最常用的 R 绘图函数,是一个泛型函数,是 R 中绘图使用最多的函数,它的函数形式如下:

```
plot(x,y,type=" ",参数设定)
```

plot( )产生的图形依赖于第一个参数的类型参数,其中 type 是指画图的类型,具体包含表 5.7 中列出的类型,默认值为"p",即散点图。

表 5.7　plot 绘图类型参数

参数	参 数 描 述
p	散点图(Points)
l	直线图(Lines)
b	点线图(Both)
c	除去点的点线图(the Lines Part Alone of"b")
o	穿过点的点线图(Both 'Overplotted')
h	直方图(Histogram)
s	阶梯图(Stairsteps)

5.2.2　饼图

饼图主要用于展示频数分布情况,在商业领域中应用非常广泛。在 R 中饼图的绘制函数为 pie(x, labels),其中 x 是一个非负数值向量,表示每个扇形的面积; labels 表示各扇形标签的字符型向量。下面举例说明。

```
> opar <- par()                              #保存当前设置
 #设置画纸为两行两列,上下边界 2 英分,左右边界 0 英分
> par(mfrow = c(2,2),mar = c(2,0,2,0))
> numbers <- c(10,12,8,9)                    #赋值
> city<-c('北京','上海','广州','深圳')        #赋值
> pie(numbers,labels = city,main = '简单饼图')  #绘制饼图
> percent <- round(numbers/sum(numbers) * 100)  #赋值
> city2 <- paste(city,'',percent,'%')        #赋值
> pie(numbers,labels = city2,col = rainbow(length(city)),
      main = '简单饼图标上百分比')             #绘制饼图,修改了标签和颜色
> library(plotrix)                           #调用 plotrix 软件包,调用之前要先安装
 #绘制三维饼图,其中 explode 为裂开的距离设置
> pie3D(numbers,labels = city,explode = 0.1,main = '三维饼图')
> attach(mtcars)                             #加载数据集
> mytable <- table(cyl)                      #获取 cyl 的频数表并赋值给 mytable
> lab <- paste(names(table(cyl)),'个气缸')    #设置标签
> pie(mytable,labels = lab,main = '用频数表绘制饼图')
> detach(mtcars)                             #释放数据集
> par(opar)                                  #恢复原来的设置
```

图 5.12 展示了饼图的一些设定,有二维的和三维的。其中,前三幅图形是假设的数据

绘制而成，第四幅图形是使用数据集 mtcars 的数据绘制的。

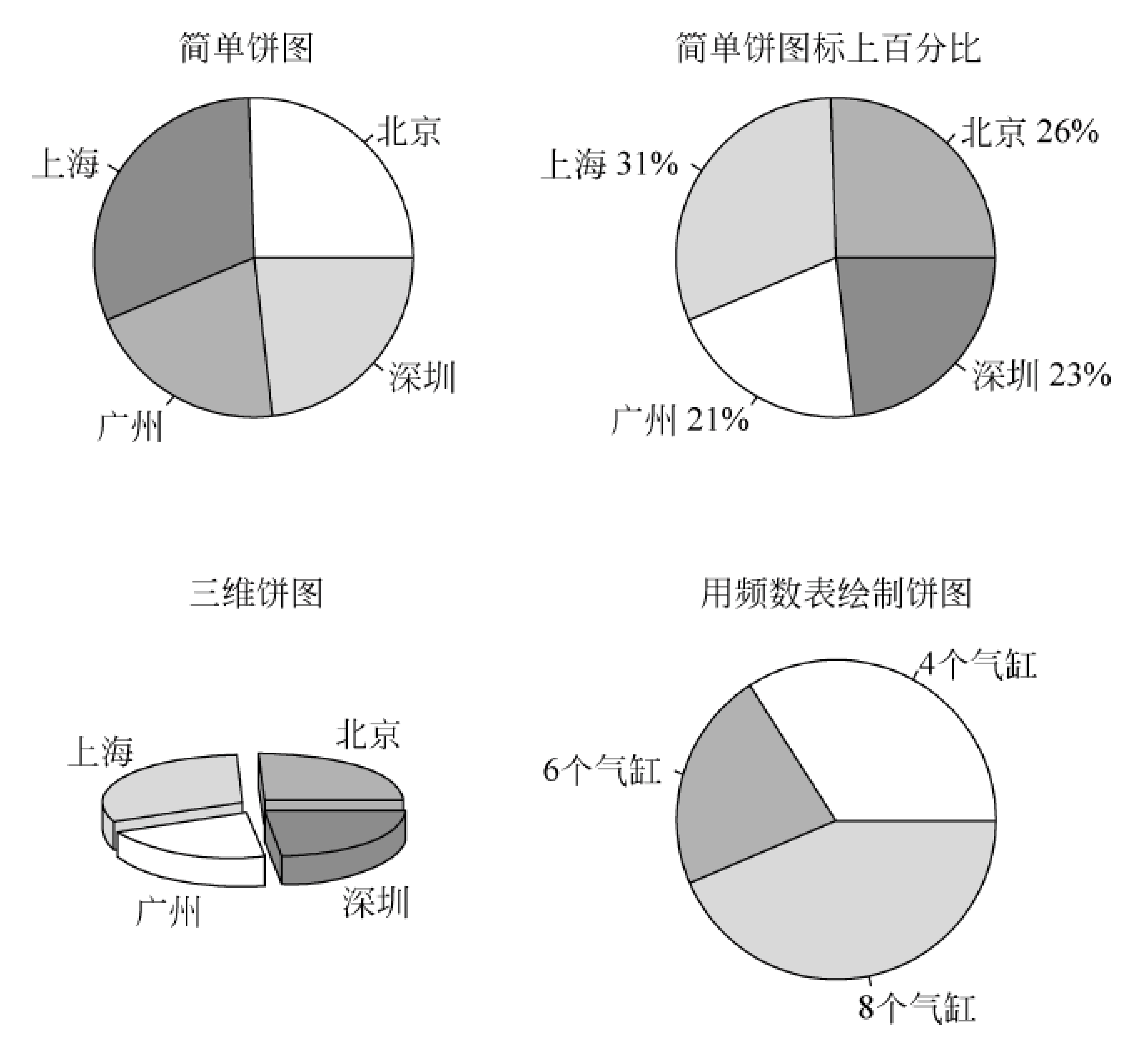


图 5.12 各种饼图

第一幅饼图是样本数据绘制的简单饼图。第二幅饼图将样本数转换为比例值，并将这项信息添加到各扇形的标签上，使用了前面绘图参数颜色中提到的 rainbow()定义了各扇形的颜色。这里的 rainbow(length(city))将被解析为 rainbow(4)，即为图形提供了 4 种颜色。第三幅饼图使用 plotrix 包中的 pie3D()创建三维饼图。第四幅饼图利用 mtcars 数据集中的 cyl(气缸数指标)先获取了它的频数表，然后利用频数表创建饼图，展示了 mtcars 数据集中的汽缸数(4 个、6 个、8 个)，并将此信息附加到了标签上，其中 8 个气缸的车所占比例最大，其次是 4 个气缸，最少的是 6 个气缸。

5.2.3 箱线图

箱线图(Boxplot)也称为箱须图(Box-whisker Plot)，是利用数据中的 5 个统计量：最小值、第一四分位数(第 25 百分位数)、中位数(第 50 百分位数)、第三四分位数(第 75 百分位数)与最大值来描述数据的一种方法。它也可以粗略地看出数据是否具有对称性，分布的分散程度等信息。箱线图能够显示出可能为离群点(范围±1.5 * IQR 以外的值，IQR 表示四分位距，即上四分位数与下四分位数的差值)的观测值，可用于几个样本的比较。箱线图函数为 boxplot(x, labels)，具体看下面的代码，图形如图 5.13 所示：

```
> attach(mtcars)
> quantile(mpg)           # 计算样本常用分位数
    0 %     25 %     50 %     75 %    100 %
 10.400  15.425  19.200  22.800  33.900
> boxplot(mpg, main = "箱线图", ylab = 'mpg')
```

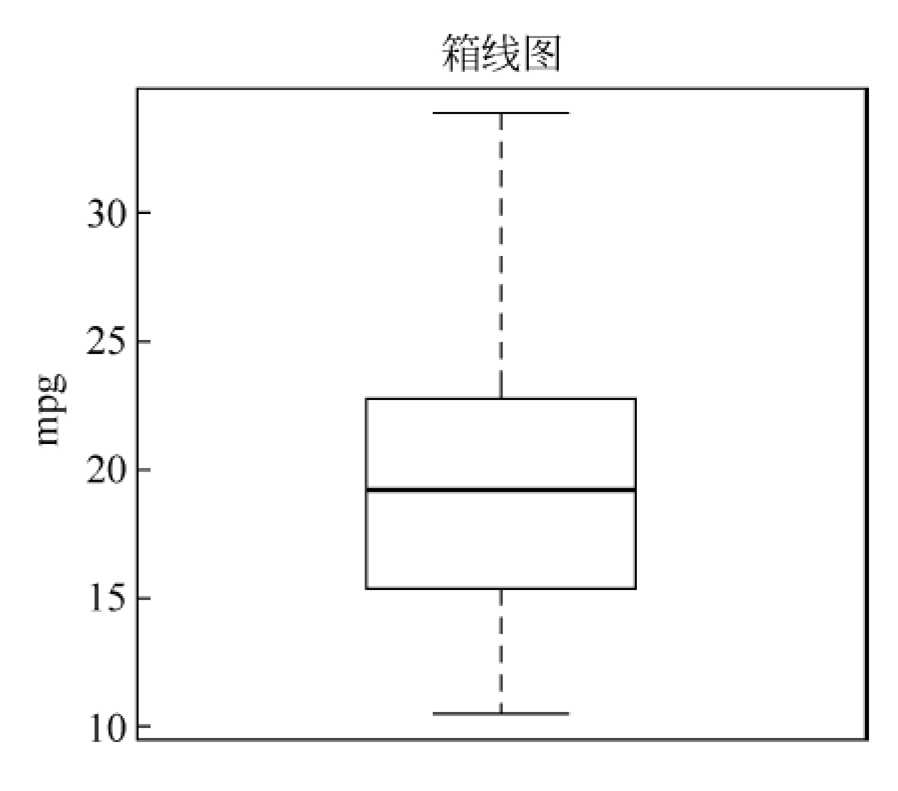


图 5.13　mtcars 数据集的箱线图

通过 quantile()的计算结果：车型样本中，每加仑汽油行驶英里数的中位数是 19.2，50%的值都落在了 15.425 和 22.8 之间，最小值为 10.4，最大值为 33.9。也可以执行 boxplot.stats(mpg)，输出用于构建箱线图的统计量。从生成的箱线图可以看出 5 点的位置，也可以大致看出频数分布情况。箱子越小，说明中间分布越集中，反之则离散程度较高。默认情况下，两条须的延伸极限不会超过盒型各端加 1.5 倍四分位距的范围。此范围以外的值将以"点"来表示，在上例样本中没有异常值。上面的须长大于下面的须长，说明分布存在右偏。

针对箱线图的对比样本的效果，使用并列箱线图进行跨组比较，函数表示为：

```
boxplot ( formula, data = dataframe )
```

其中，formula 是一个公式，dataframe 代表提供数据的数据框(或列表)。一个示例公式为 y～A，这将为变量 A 的每个值并列地生成数值型变量 y 的箱线图。公式 y～A * B 则将为变量 A 和 B 所有水平的两两组合生成数值型变量 y 的箱线图。添加参数 varwidth=TRUE，将使箱线图的宽度与其样本大小的平方根成正比，参数 horizontal=TRUE 可以反转坐标轴的方向。

1. 单个因子的箱线图

使用并列箱线图继续研究 4 个气缸、6 个气缸、8 个气缸发动机对每加仑汽油行驶的英里数的影响，并进行分类汇总，分别绘制描述不同气缸的汽车 mpg 分布的箱线图，结果如图 5.14 所示，代码如下：

```
> boxplot(mpg ~ cyl, data = mtcars, main = "汽车里程数据",
    col = c(5,6,7), xlab = "cyl", ylab = "mpg")   # 按照 cyl 对 mpg 进行分类绘制箱线图
```

由图 5.14 可以看到不同组间油耗的区别非常明显，气缸越多，耗油量越大；6 个气缸车型的每加仑汽油行驶的英里数分布较其他两类车型更为均匀。与其他两个气缸车型相比，4 个气缸车型的每加仑汽油行驶的英里数散布最广且存在正偏，在 8 个气缸组还有一个离群点。

2. 两个交叉因子的箱线图

下面考虑不同气缸数和不同变速箱类型的车型，两种因子交叉影响下的箱线图如

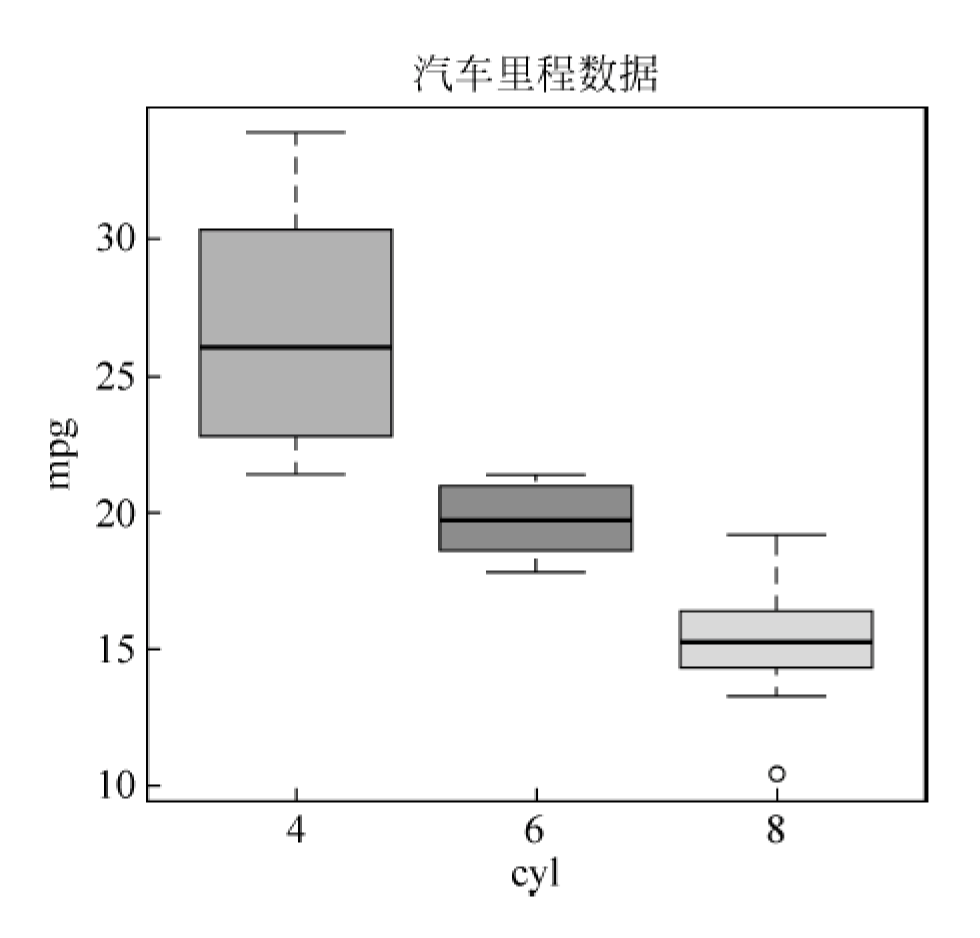


图 5.14　多样本箱线图

图 5.15 所示，具体代码如下：

```
#按照类别型变量 cyl 和 am 所有水平的两两组合生成数值型变量 mpg 的箱线图，并使箱线图的宽度与其样本大小的平方根成正比
>cyl.f<-factor(cyl,levels=c(4,6,8), labels=c("4","6","8"))      #定义因子 cyl.f
>am.f<-factor(am,levels=c(0,1), labels=c("自动挡","手动挡"))   #定义因子 am.f
#绘制双因子箱线
>boxplot(mpg~am.f*cyl.f,data=mtcars, varwidth = T, col = c(5,6),
        main = "mpg 按照汽车类型分布", xlab = "汽车类型")
```

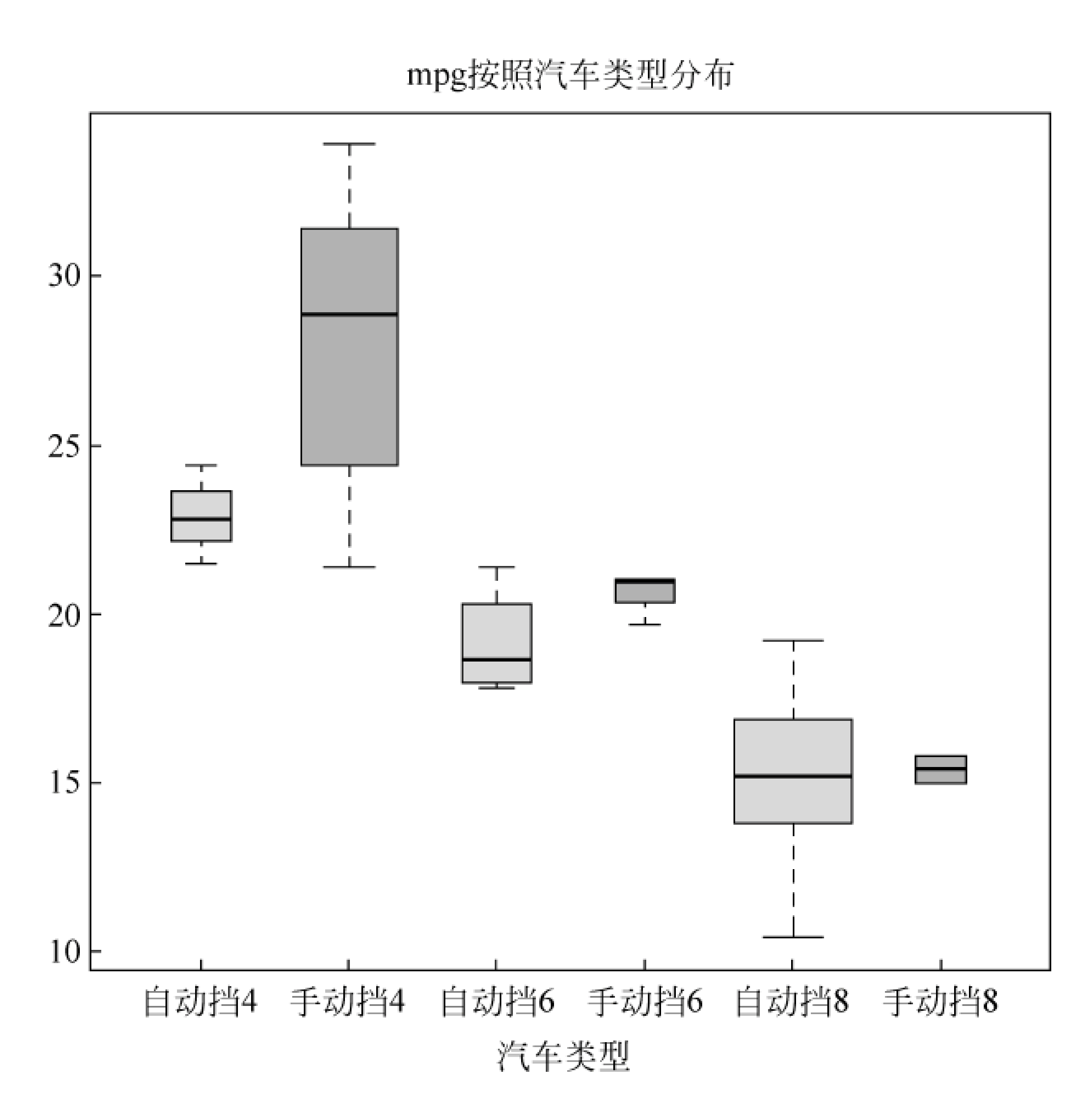


图 5.15　两个交叉因子的箱线图

图 5.15 绘制了每加仑汽油行驶英里数按照汽车类型分布的箱线图。同样的，这里使用参数 col 为箱线图进行着色，请注意颜色的循环使用。在本例中共有 6 幅箱线图和两种指定的颜色，所以颜色将重复使用三次。

图 5.15 清晰地显示出油耗随着缸数的增多而变大。对于 4 个气缸和 6 个气缸车型，标准变速箱(手动挡)的油耗更高；但对于 8 个气缸车型，油耗似乎没有差别。从箱线图的宽度可以看出，4 个气缸标准变速箱的车型和 8 个气缸自动变速箱的车型在数据集中最常见。

5.2.4　条形图

条形图通过垂直的或水平的条形展示了类别型变量的分布(频数)，函数为 barplot (x)，其中 x 是一个向量或一个矩阵。

(1) 若 x 是一个向量，则它的值就确定了各条形的高度，并将绘制一幅垂直的条形图。

(2) 若 x 是一个矩阵而不是向量，那么绘图结果将是一幅堆砌条形图或分组条形图。如果 beside＝FALSE(默认值)，那么矩阵中的每一列都将生成图中的一个条形，各列中的值将给出堆砌的“子条”的高度。若 beside＝TRUE，那么矩阵中的每一列都表示一个分组，各列中的值将并列而不是堆砌。

另外，使用选项 horiz＝TRUE 则会生成一幅水平条形图。同时可以添加标注选项，其中 main 选项可添加一个图形标题，而 xlab 和 ylab 选项则会分别添加 x 轴和 y 轴标签，具体参考下面的例子。

1. x 是一个向量

```
> attach(mtcars)
 #绘制简单条形图(如图 5.16 所示)
> gear.c <- table(gear)
> barplot(gear.c,main = "Car Distribution",,col = c(5,6,7)
xlab = "Number of Gears")
 #水平放置,设置标签(如图 5.17 所示)
> barplot(gear.c, main = "Car Distribution",horiz = T,
            names.arg = c("3 Gears", "4 Gears", "5 Gears"),col = c(5,6,7))
```

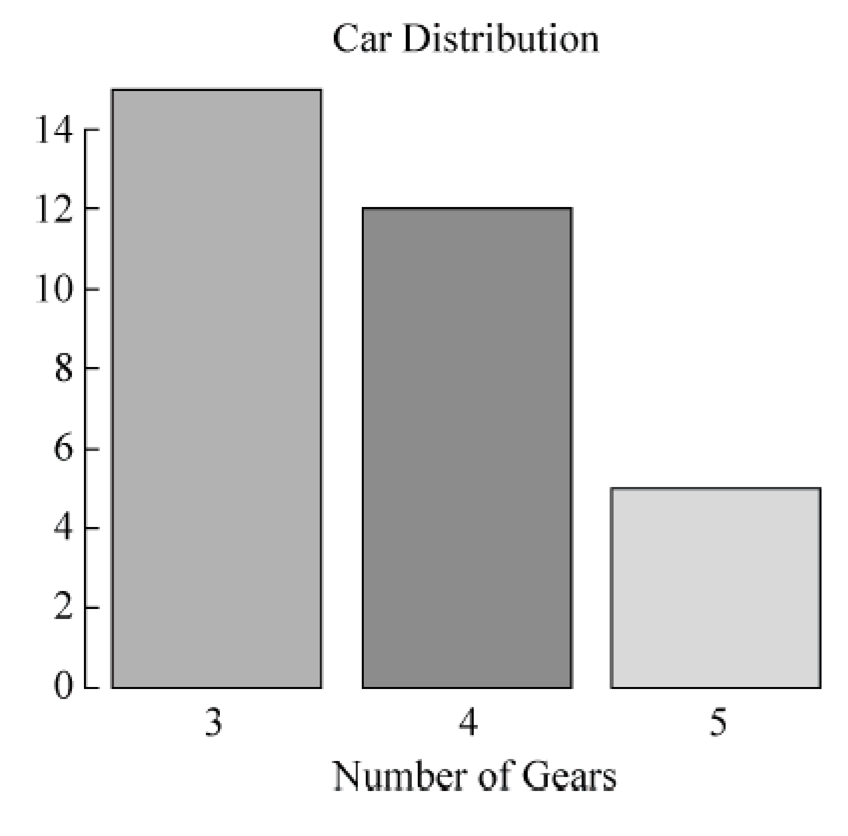


图 5.16　向量条形图纵向演示

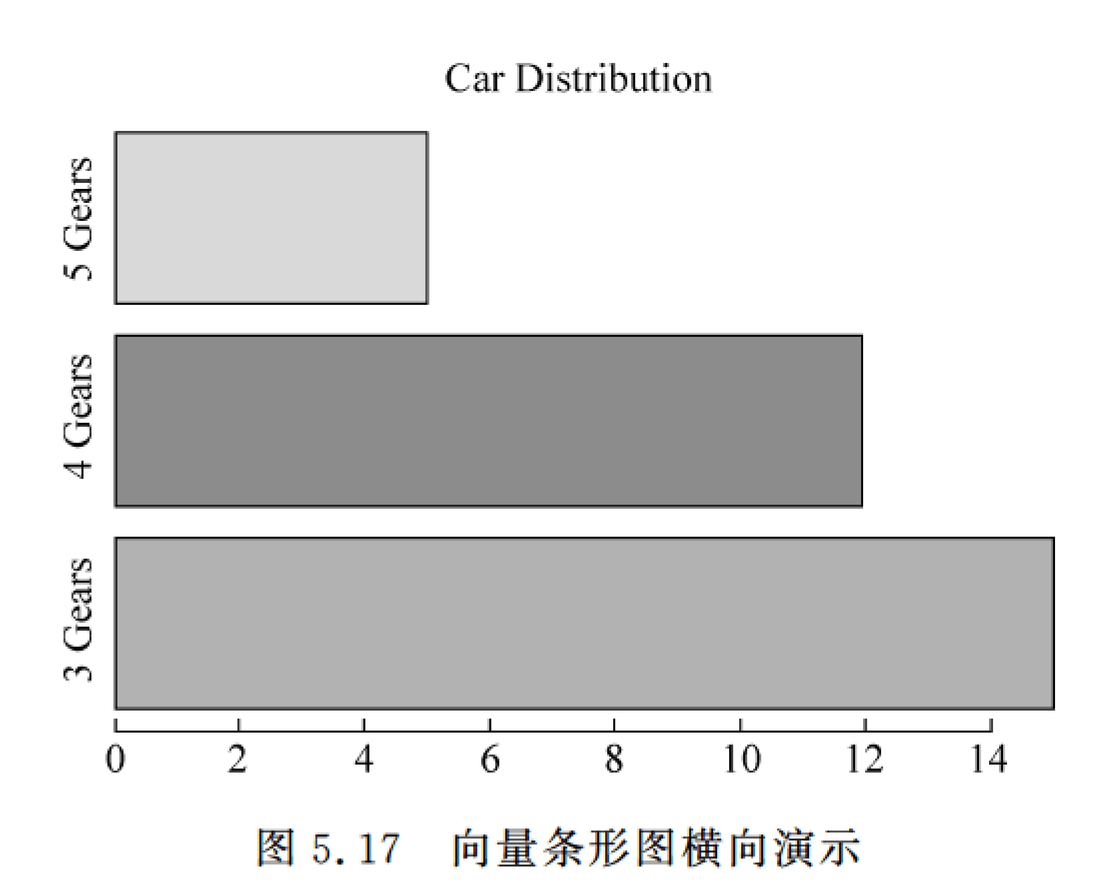


图 5.17　向量条形图横向演示

2. x 是一个矩阵

```
#堆积条形图,设置填充颜色和图例(如图 5.18 所示)
>vg.c <- table(vs, gear)
>barplot(vg.c,main = "Car Distribution by Gears and VS",
        xlab = "Number of Gears", col = c("darkblue","red"),
        legend = rownames(vg.c))
#分组条形图(如图 5.19 所示)
>barplot(vg.c,main = "Car Distribution by Gears and VS (2)",
          xlab = "Number of Gears",col = c("darkblue","red"),
          legend = rownames(counts),beside = T)
>detach(mtcars)
```

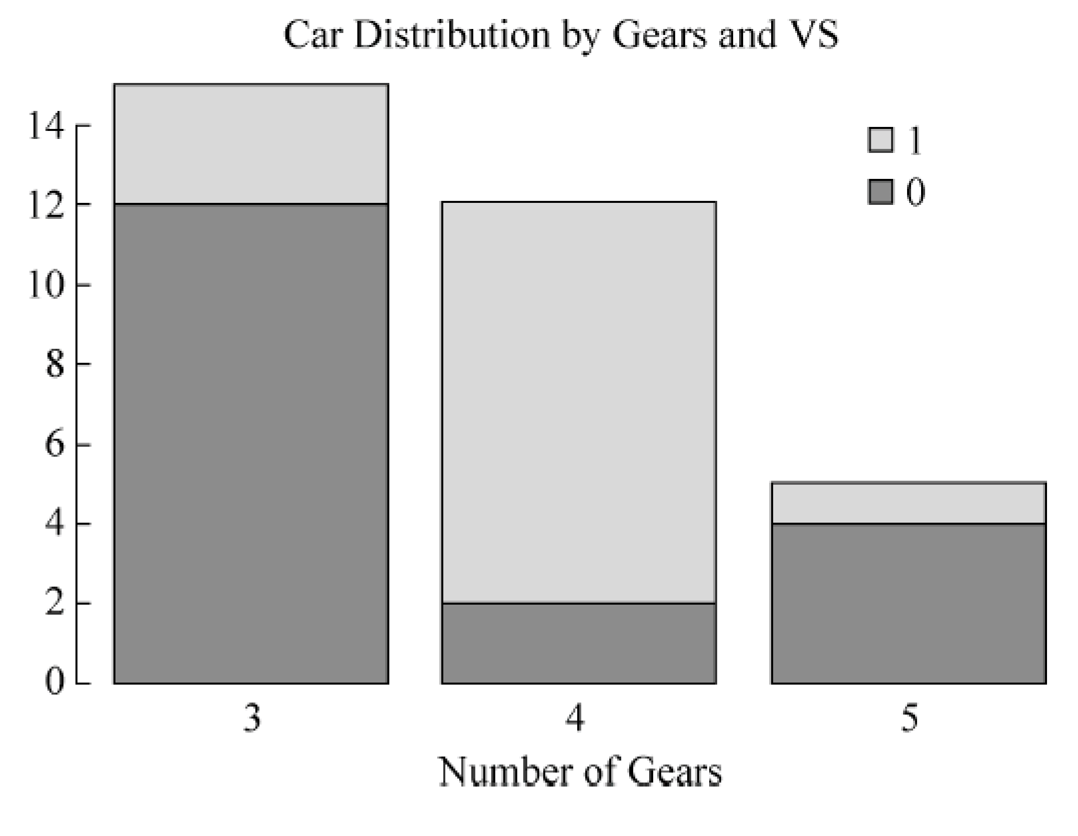


图 5.18　堆积条形图

5.2.5　直方图

直方图通过在 x 轴上将值域分割为一定数量的组,在 y 轴上显示相应值的频数,展示了连续型变量的分布,可以使用 hist(x)创建直方图,其中 x 是一个由数据值组成的数值向量。参数 freq=FALSE 表示根据概率密度而不是频数绘制图形,参数 breaks 用于控制组的数

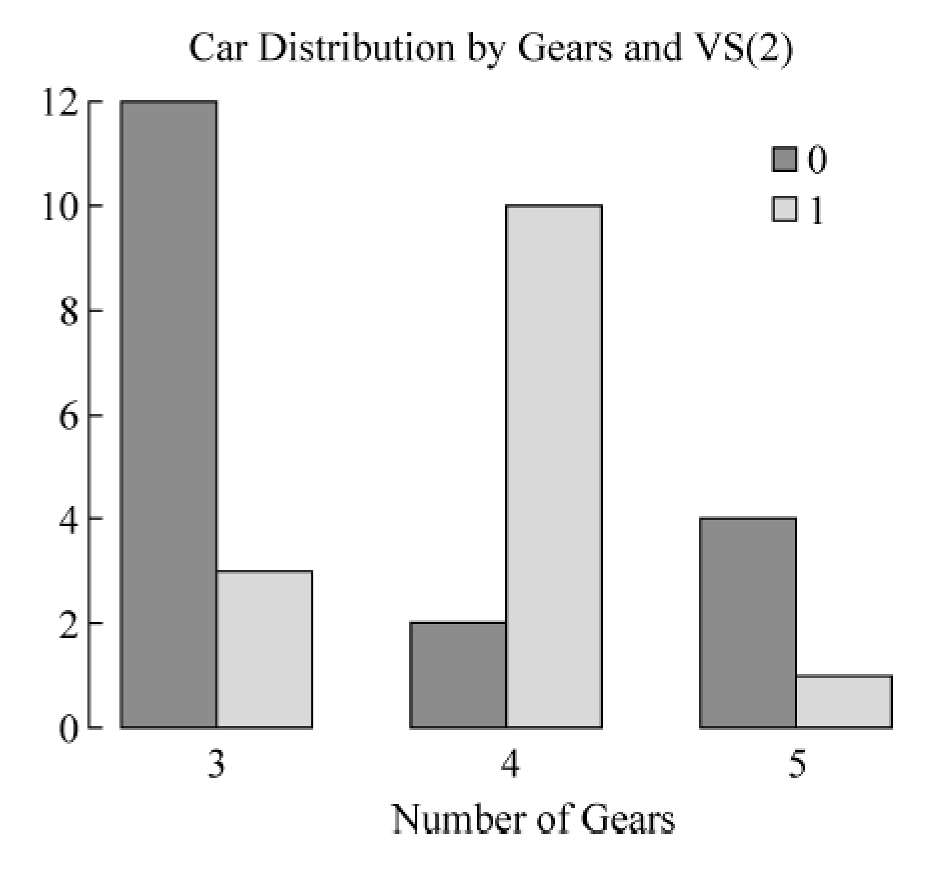
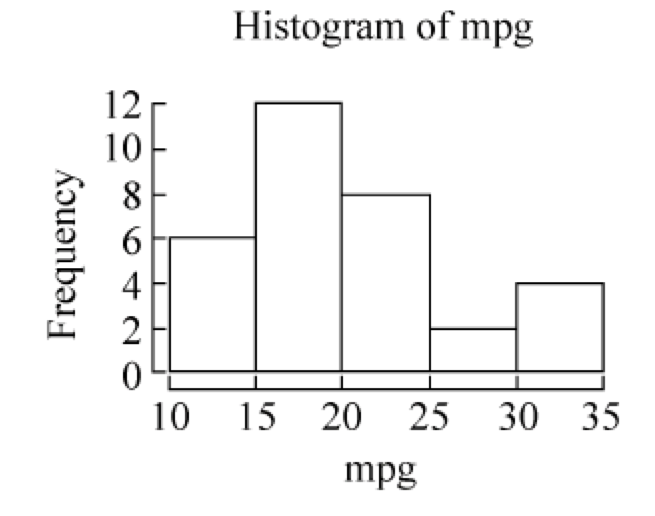


图 5.19　分组条形图

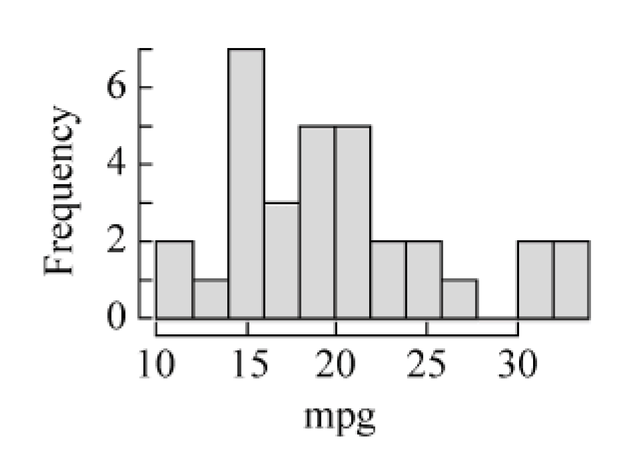
量。在定义直方图中的单元时，默认将生成等距切分，具体代码如下，图形如图 5.20 所示：

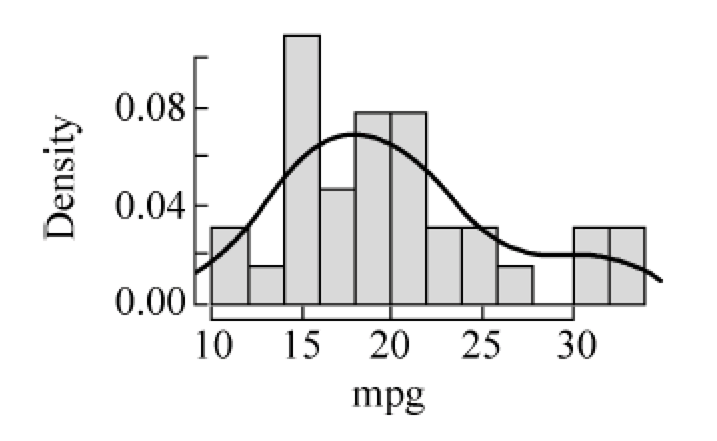
```
> opar = par()                    # 默认设置赋值给 opar
> attach(mtcars)                  # 加载数据集 mtcars
> par(mfrow = c(2, 2))            # 设置为 4 幅图的画纸
 # 绘制第一幅：mpg 直方图，其他为默认设置
> hist(mpg)
 # 绘制第二幅：有 12 组的红色 mpg 直方图
> hist(mpg, breaks = 12, col = "red", xlab = "mpg",
     main = "Colored histogram with 12 bins")
 # 绘制第三幅：按照概率密度绘制的直方图，添加蓝色概率密度曲线
> hist(mpg, freq = F, breaks = 12, col = "red", xlab = "mpg",
       main = "Histogram,density curve")
> lines(density(mpg), col = "blue", lwd = 2)
 # 绘制第四幅：直方图加上拟合的正态曲线和边框
> x <- mpg
> h <- hist(x, breaks = 12, col = "red",xlab = "mpg",
              main = "Histogram with normal curve and box")
> xfit <- seq(min(x), max(x), length = 40)
> yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
> yfit <- yfit * diff(h$mids[1:2]) * length(x)
> lines(xfit, yfit, col = "blue", lwd = 2)
> box()
> par(opar)                       # 恢复默认设置
> detach(mtcars)                  # 释放数据集 mtcars
```

第一幅直方图展示了未指定任何选项时的默认图形，自动生成了 5 个组，并且显示了默认的标题和坐标轴标签。第二幅直方图将组数指定为 12，使用红色填充条形，并添加了更吸引人、更具信息量的标签和标题。第三幅直方图保留了上一幅图中的颜色、组数、标签和标题设置，添加了一条密度曲线，它为数据的分布提供了一种更加平滑的描述。第四幅直方图在第二幅直方图的基础上加入了一条正态曲线。









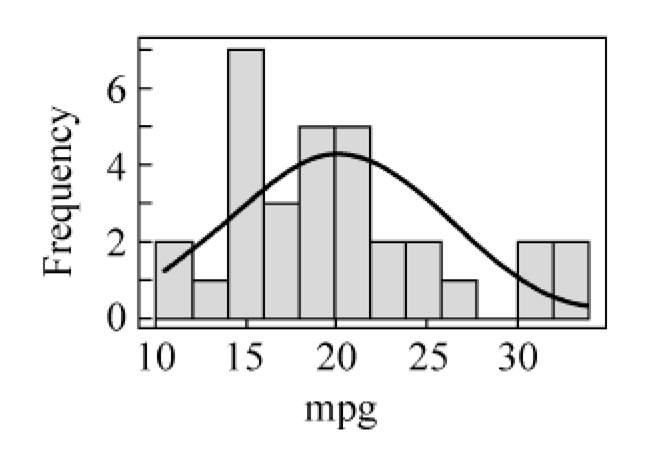


图 5.20　直方图相关演示

5.2.6　核密度图

核密度估计(Kernel Density Estimation)是在概率论中用来估计未知的密度函数,属于非参数检验方法之一。核密度估计方法不利用有关数据分布的先验知识,对数据分布不附加任何假定,是一种从数据样本本身出发研究数据分布特征的方法。核密度图是一种用来观察连续型变量分布的有效方法,绘制密度图的语法为 plot(density(x)),其中 x 是一个数值型向量。由于 plot()会创建一幅新的图形,因此要向一幅已经存在的图形上叠加一条密度曲线可以使用 lines()(如上一小节直方图绘制中所示)。具体代码如下,图形如图 5.21 所示:

```
>attach(mtcars)
>par(mfrow = c(2, 1))
>mpg.d <- density(mpg)
>plot(mpg.d)                          #第一幅,核密度图
>plot(mpg.d, main = "mpg 核密度图")#第二幅,着色的核密度图
>polygon(mpg.d, col = "red", border = "blue")
>rug(mtcars$mpg,col = "brown")
>par(opar)                            #恢复默认设置
>detach(mtcars)
```

在第一幅图中使用默认设置创建的最简图形。在第二幅图中添加了一个标题,将曲线修改为蓝色,使用实心红色填充了曲线下方的区域,并添加了棕色的轴须图。polygon()根据顶点的 x 和 y 坐标(本例中由 density()提供)绘制了多边形。

核密度图可用于比较组间差异。可能是由于普遍缺乏方便好用的软件,这种方法其实没有被充分利用。幸运的是,sm 包填补了这一缺口,其中的 sm.density.compare( )可向图形叠加两组或更多的核密度图,其语法格式为 sm.density.compare(x, factor),其中 x 是一个数值型向量,factor 是一个分组变量。利用 mtcars 数据集进行列举,比较分别拥有 4 个、6

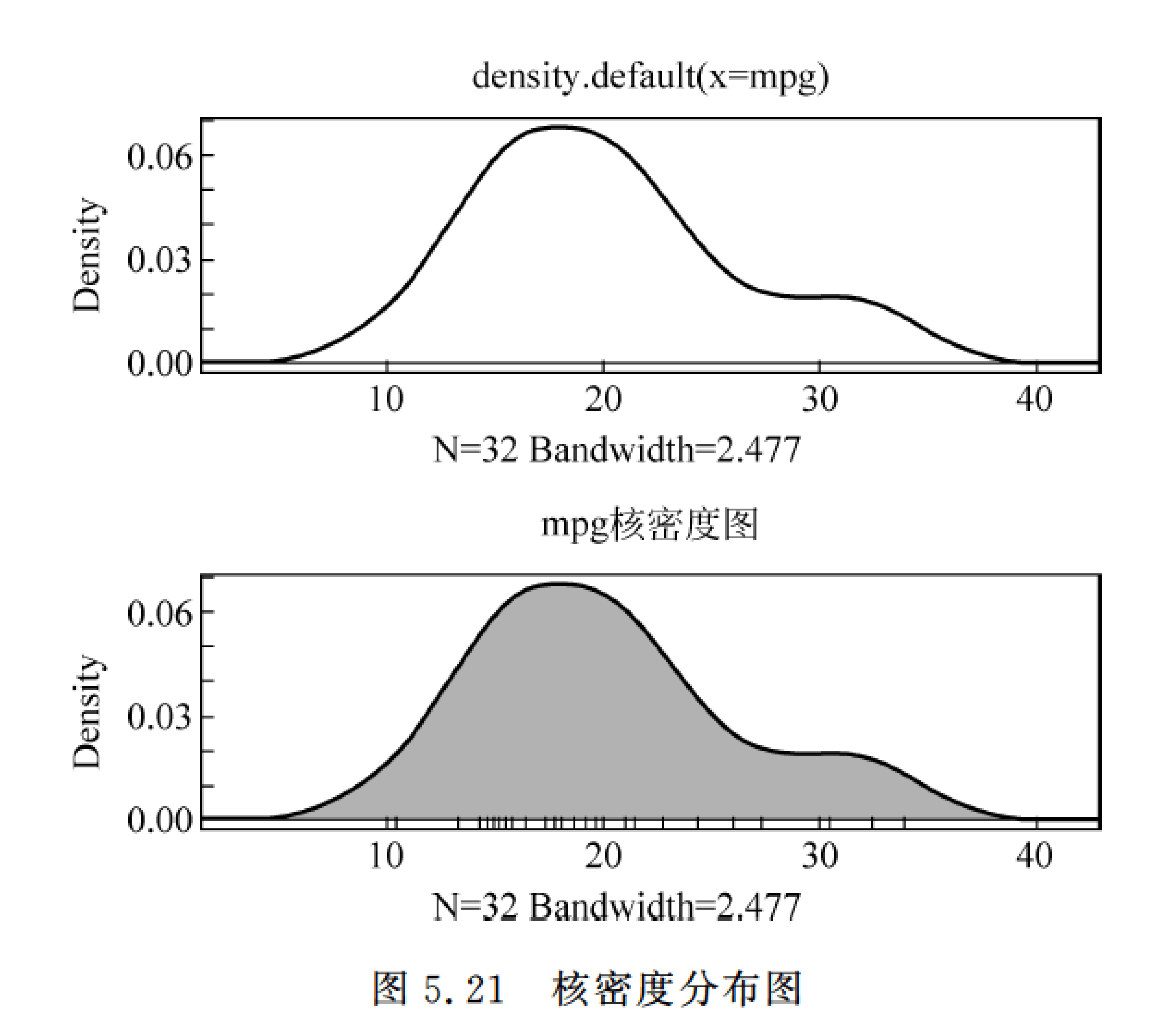


图 5.21　核密度分布图

个或 8 个汽缸车型的每加仑汽油行驶英里数，代码如下：

```
> library(sm)
> attach(mtcars)
> cyl.f <- factor(cyl, levels = c(4, 6, 8), labels = c("4 cylinder", "6 cylinder",
           "8 cylinder"))                    # 定义分类因子
# 绘制核密度图
> sm.density.compare(mpg, cyl, xlab = "Miles Per Gallon")
> title(main = "不同汽缸数的 mpg")
> colfill <- c(2:(2 + length(levels(cyl.f))))  # 设定填充颜色
# 添加图例
> legend(locator(1), levels(cyl.f), fill = colfill)
> detach(mtcars)
```

图 5.22 中的图例增加了图表的可解释性。上述代码首先创建的是一个颜色向量，这里

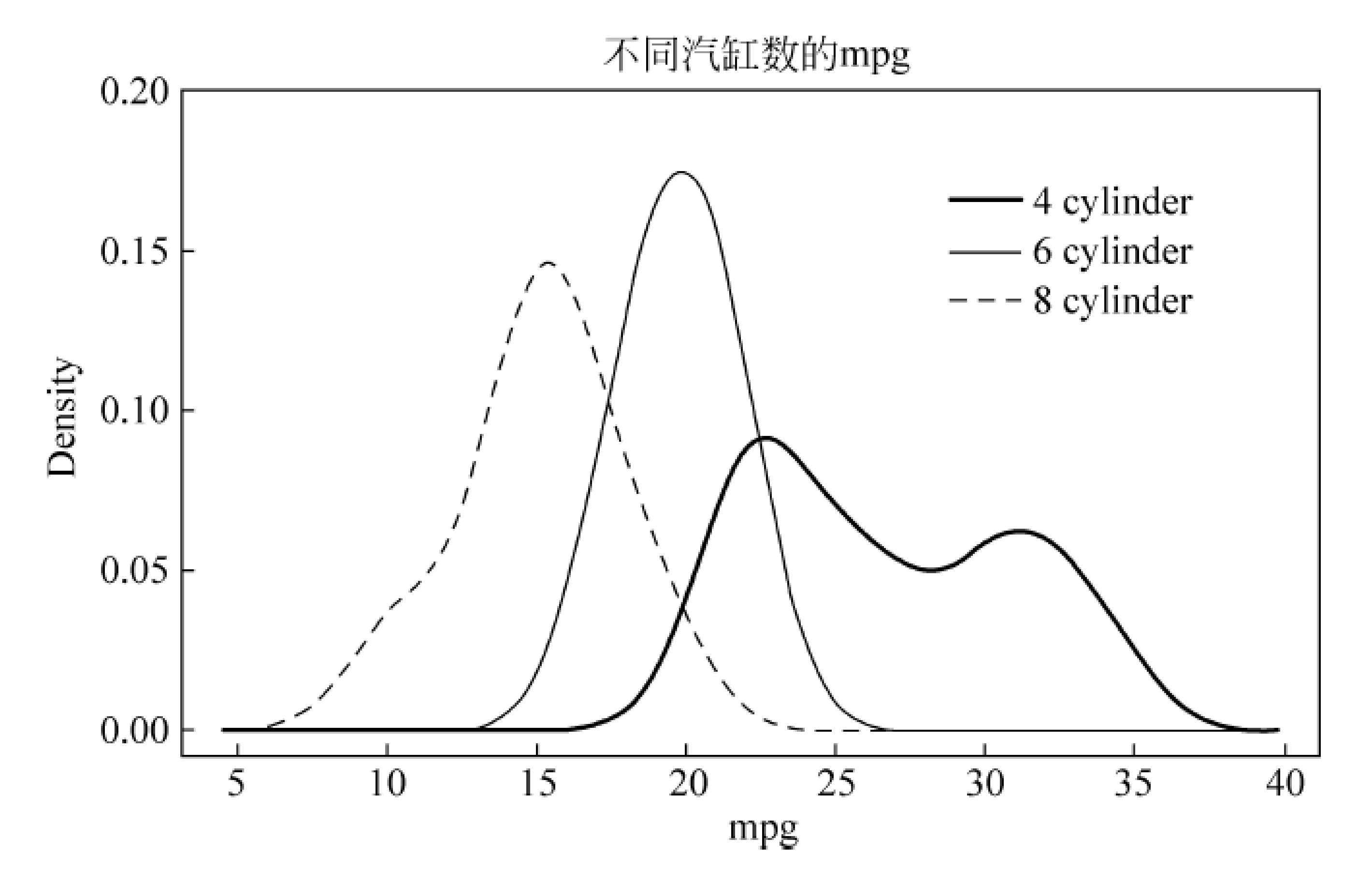


图 5.22　各汽缸数的核密度曲线

的 colfill 值为 c(2, 3, 4)，然后通过 legend()向图形上添加一个图例。第一个参数值 locator(1)表示用鼠标单击想让图例出现的位置来交互式地放置这个图例；第二个参数值则是由标签组成的字符向量；第三个参数值使用向量 colfill 为 cyl.f 的每一个水平指定了一种颜色。

如图 5.22 所示，核密度图的叠加不失为一种在某个结果变量上跨组比较观测的强大方法，可以看到不同组所含值的分布形状，以及不同组之间的重叠程度不同。

5.2.7 点图

点图给出了一种在水平刻度线上创建有大量标签值的方法，点图函数为 dotchart(x, labels=)，其中 x 为一个数值向量，labels 表示每个点的标签构成的向量。下面运用该函数绘制点图，R 代码如下，图形如图 5.23 所示：

```
> attach(mtcars)
> dotchart(mtcars $ mpg, labels = row.names(mtcars),
 +  main = "mpg 点图")                          # 点图
> par(opar)                                     # 恢复默认设置
> detach(mtcars)
```

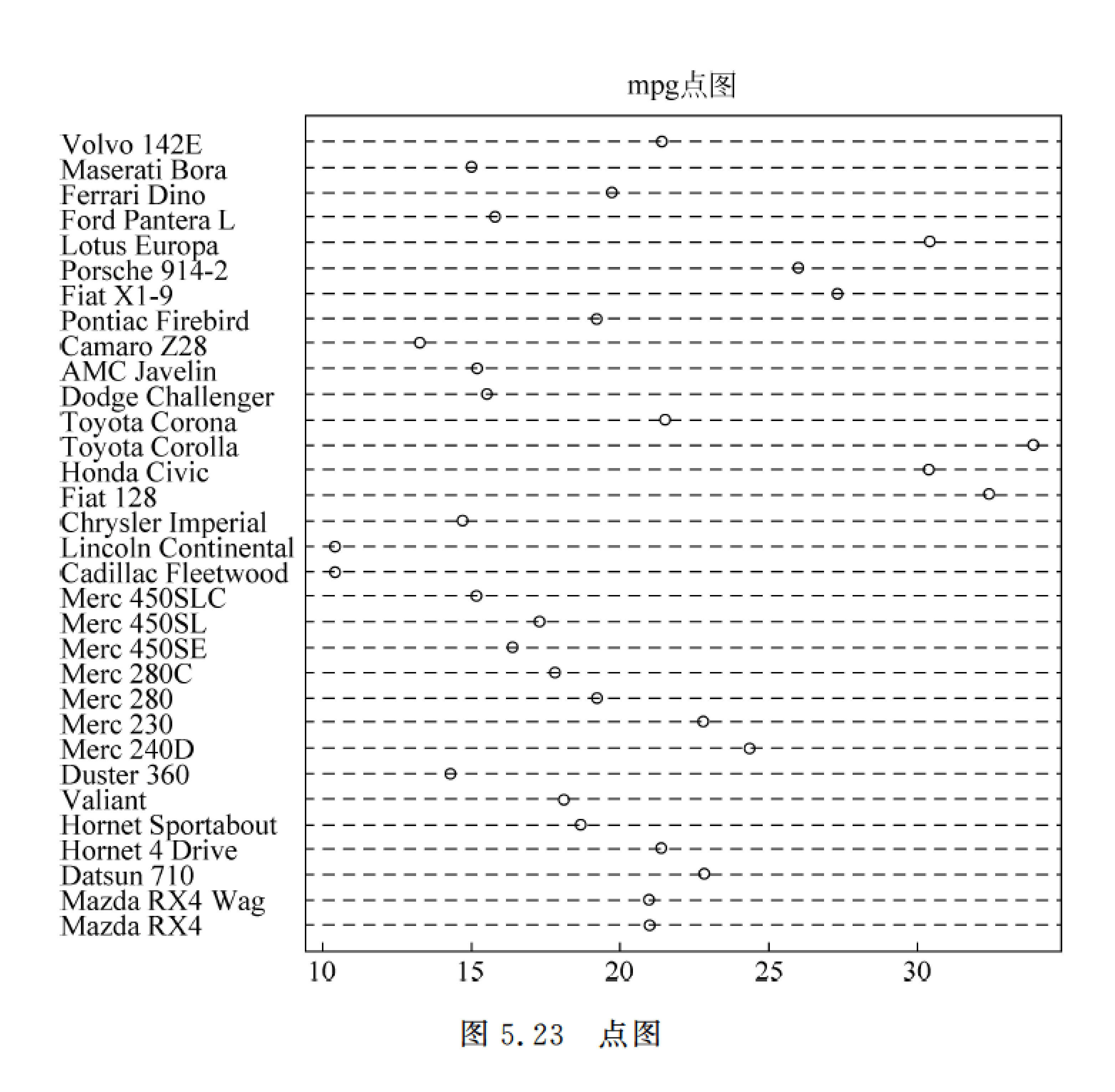


图 5.23 点图

可以使用 color 设置点图的颜色，具体代码如下，图形如图 5.24 所示。

```
> mtcars $ cyl <-  factor(mtcars $ cyl)
> mtcars $ color[mtcars $ cyl == 4] <-  "blue"
> mtcars $ color[mtcars $ cyl == 6] <-  "red"
> mtcars $ color[mtcars $ cyl == 8] <-  "brown"
> dotchart(mtcars $ mpg, labels = row.names(mtcars), color = mtcars $ color,
 +  main = "mpg 点图")                  # 第二幅，着色的点图
> par(opar)                           # 恢复默认设置
> detach(mtcars)
```



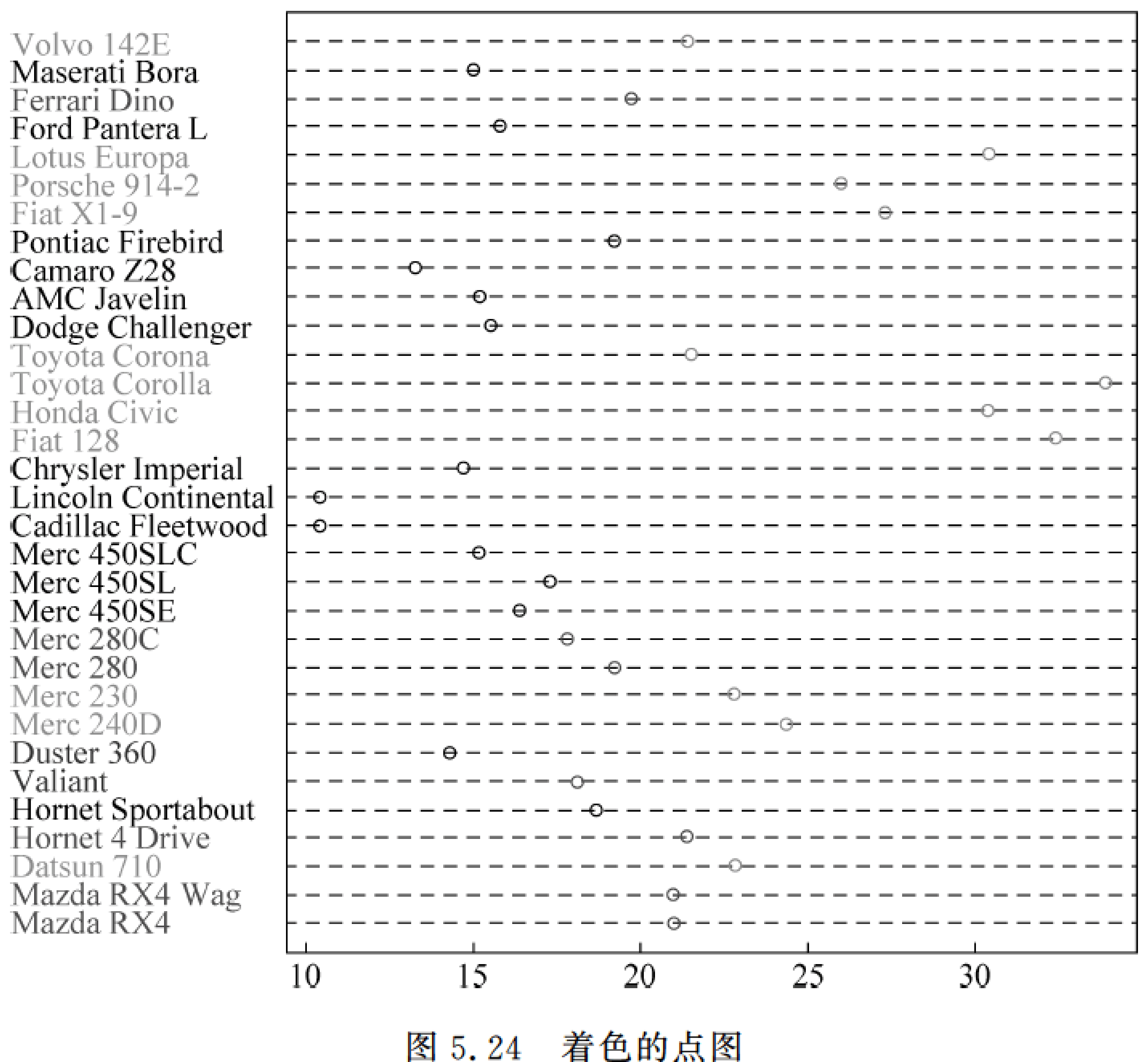


图 5.24　着色的点图

5.3　低级绘图函数

低级绘图函数是指在高级绘图函数绘制出的图上进行点、直线、线段、箭头、网格线等的添加，使图更加丰富。表 5.8 列出了部分常用的低级绘图函数。

表 5.8　低级绘图函数表

函 数 名 称	函 数 描 述
points	在当前绘图区增加点
lines	在当前绘图区增加连接线
abline(a,b)	在当前绘图区增加一个斜率为 b，截距为 a 的直线
abline(h=y)	h=y 可用于指定贯穿整个图的水平线高度的 y 坐标

续表

函 数 名 称	函 数 描 述
abline(v=x)	v=x 类似地用于指定垂直线的 x 坐标
abline(lm:obj)	lm:obj 可能是一个有长度为 2 的 coefficients 分量(如模型拟合的结果)的列表，该分量中依次含有截距和斜率
segments	绘制点对之间的线段
arrows	绘制点对之间的箭头
grid	在当前绘图区增加网格线

这里只进行简单的举例，不再进行太多的赘述，代码如下，图形如图 5.25 所示：

```
plot( - 4:4,  - 4:4, type  =  "p", col = "blue")                        #基本实现
points(x = c(3, - 2, - 1,3,2), y = c(1,2, - 2,2,3), col  =  "red")      #绘制点、连接点
lines(x = c(3, - 2, - 1,3,2), y = c(1,2, - 2,2,3),col = "black")        #绘制直线
abline(h = 0) ; abline(v = 0)
abline(a = 1,b = 1)
abline(lm(mtcars $ mpg ～ mtcars $ qsec),col = "red")
segments(x0 = 2, y0 =  - 4.5, x1 = 4, y1 =  - 2, col = "red", lty = "dotted")  #绘制线段
arrows(x0 =  - 4, y0 = 4, x1 =  - 2, y1 = 0, length = 0.15, angle = 30, code = 3)
grid(nx = 3, ny = 5, col  =  "lightgray", lty  =  "dotted")               #绘制网格线
```

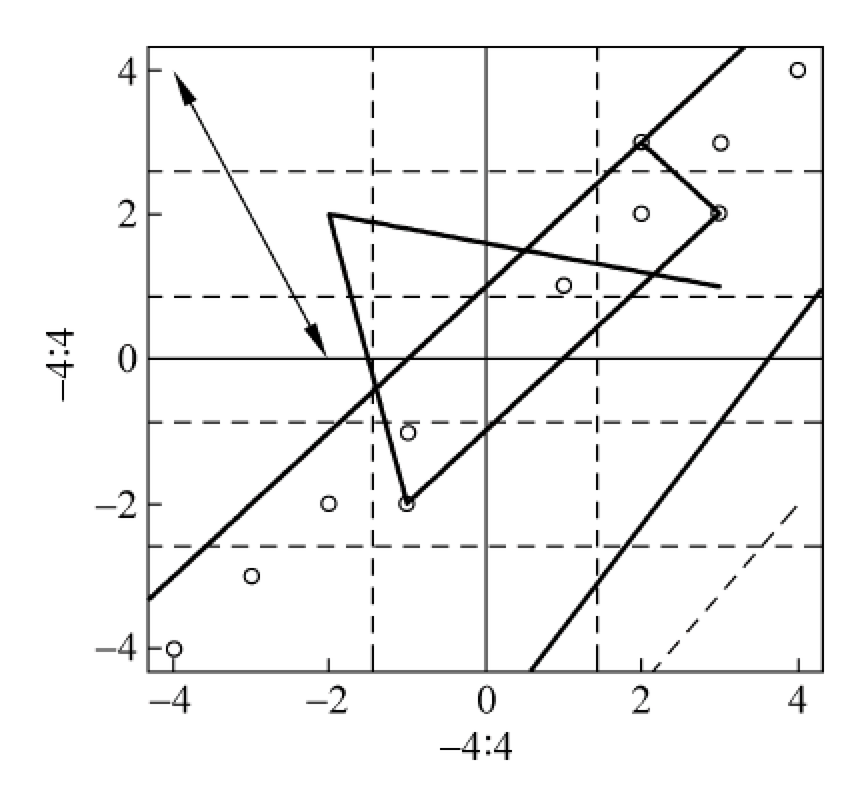


图 5.25　低级函数绘图

CHAPTER 6

第6章 R语言数据分析

数据分析是 R 的重要功能,包括基本的数据处理函数、多元统计分析的实现等。本章主要介绍 R 进行数据处理常用的数学统计等基础函数,以及方差分析、判别分析、聚类分析、主成分分析、因子分析和典型相关分析等多元统计分析方法。

6.1 数据处理基础函数

本节将综述 R 中作为数据处理基础的函数,它们可分为数值(数学、统计、概率)函数和字符处理函数。

6.1.1 数学函数

在一些函数或模型中常常遇到一些数学类的计算。对此,R 也给出了相应的计算函数,表 6.1 列出了一些 R 常用数学函数。

表 6.1 R 常用数学函数

函数	描述
abs(x)	绝对值,如 abs(-4)的返回值为 4
sqrt(x)	平方根,如 sqrt(25)的返回值为 5
ceiling(x)	向上取整,如 ceiling(3.475)的返回值为 4;ceiling(-3.475)的返回值为-3
floor(x)	向下取整,如 floor(3.475)的返回值为 3;floor(-3.475)的返回值为-4
trunc(x)	截取整数部分,如 trunc(5.99)的返回值为 5;trunc(-5.99)的返回值为-5
round(x,digits=n)	四舍五入,指定小数的位数,如 round(3.475, digits=2)的返回值为 3.48
log_n(x)(x,base=n)	对 x 取以 n 为底的对数
log(x)	自然对数,如 log(10)的返回值为 2.3026
log10(x)	常用对数,如 log10(10)的返回值为 1
exp(x)	指数函数,如 exp(2.3026)的返回值为 10

6.1.2 统计函数

R 的统计计算功能强大，在数据分析、建模时往往需要借助一些基础的统计函数对数据进行简单的统计分析，表 6.2 列出了一些 R 常用的统计函数。

表 6.2 R 常用的统计函数

函　　数	描　　述
mean(x)	平均数，如 mean(c(1,2,3,4))的返回值为 2.5
median(x)	中位数，如 median(c(1,2,3,4))的返回值为 2.5
sd(x)	标准差，如 sd(c(1,2,3,4))的返回值为 1.29
var(x)	方差，如 var(c(1,2,3,4))的返回值为 1.67
mad(x)	绝对中位差(Median Absolute Deviation)，如 mad(c(1,2,3,4))的返回值为 1.48
quantile(x, probs)	求分位数，其中 x 为待求分位数的数值型向量，probs 为一个由[0,1]之间的概率值组成的数值向量，如： 求 x 的 30%和 84%分位点 y<-quantile(x, c(.3,.84))
range(x)	求值域，如 x<-c(1,2,3,4)，range(x)的返回值为 c(1,4)，diff(range(x))的返回值为 3
sum(x)	求和，如 sum(c(1,2,3,4))的返回值为 10
diff(x,lag=n)	滞后差分，lag 用以指定滞后几项。默认的 lag 值为 1，如： x<-c(1, 5, 23, 29) diff(x)的返回值为 c(4, 18, 6)
min(x)	求最小值，如 min(c(1,2,3,4))的返回值为 1
max(x)	求最大值，如 max(c(1,2,3,4))的返回值为 4
scale(x, center = T, scale =T)	为数据对象 x 按列进行中心化(center=TRUE)或标准化(center=TRUE, scale =TRUE)

下面具体介绍两类常用的函数。

1. scale()

scale()用于对数据进行标准化处理，默认情况下 scale()对矩阵或数据框的指定列进行均值为 0、标准差为 1 的标准化处理，命令如下：

```
newdata <- scale(mydata)
```

要对每一列进行任意均值和标准差的标准化处理，命令如下：

```
newdata <- scale(mydata) * SD + M
```

其中，M 表示指定的均值；SD 表示指定的标准差。在非数值型的列上使用 scale()将会报错。要对指定列而不是整个矩阵或数据框进行标准化，可以使用如下命令：

```
newdata <- transform(mydata, myvar = scale(myvar) * SD + M)
```

该命令将变量 myvar 标准化为均值 M 、标准差为 SD 的变量。

2. apply()

```
apply(A, MARGIN, FUN, … )
```

其中,A 为数据对象,MARGIN 表示维度的下标,FUN 为指定的函数,“…”包括了任何想传递给 FUN 的参数。在矩阵或数据框中,MARGIN=1 表示行,MARGIN=2 表示列。

6.1.3　概率函数

在进行数据分析时经常会遇到各种分布的假设和检验,R 提供了大量的概率函数可以直接调用,具体如表 6.3 所示。

表 6.3　概率函数

分布名称	R 函数	分布名称	R 函数
Beta 分布	beta	Logistic 分布	logis
二项分布	binom	多项分布	multinom
柯西分布	cauchy	负二项分布	nbinom
卡方分布(非中心)	chisq	正态分布	norm
指数分布	exp	泊松分布	pois
F 分布	f	Wilcoxon 符号秩分布	signrank
Gamma 分布	gamma	*t* 分布	t
几何分布	geom	均匀分布	unif
超几何分布	hyper	Weibull 分布	weibull
对数正态分布	lnorm	Wilcoxon 秩和分布	wilcox

在 R 中对于不同分布在求密度函数、分布函数、分位数函数和生成随机数时,依次使用 d、p、q、r 后面跟上相应的函数名称即可。比如,对于标准正态分布(均值为 0,标准差为 1)而言,其密度函数为 dnorm、分布函数为 pnorm、分位数函数为 qnorm、随机数生成函数为 rnorm。总而言之,只需在概率分布前加上 d、p、q、r 这 4 项前缀就可以实现该分布的相应效果,具体可以通过 help()进行了解。

以正态分布为例,如果不指定一个均值和一个标准差,则有:

(1) d =密度函数(density)

dnorm (x,…),其中 x 为数值向量。

(2) p =分布函数(distribution function)

pnorm (q,…),其中 q 为数值向量。

(3) q =分位数函数(quantile function)

qnorm (p,…),其中 p 为概率构成的数值向量。

(4) r=生成随机数(random)

rnorm (n,…),其中 n 为生成数据的个数。

6.1.4　数据分析实例

1. 问题分析

一组学生参加了数学、科学和英语考试。为了给所有学生确定一个单一的成绩衡量指

标，需要将这些科目的成绩组合起来，将前 20% 的学生评定为 A，接下来 20% 的学生评定为 B，以此类推，并且按字母顺序对学生排序。数据如表 6.4 所示。

表 6.4 学生的考试成绩

Student	Math	Science	English
John Davis	502	95	25
Angela Williams	600	99	22
Bullwinkle Moose	412	80	18
David Jones	358	82	15
Janice Markhammer	495	75	20
Cheryl Cushing	512	85	28
Reuven Ytzrhak	410	80	15
Greg Knox	625	95	30
Joel England	573	89	27
Mary Rayburn	522	86	18

观察此数据集，可以发现一些明显的问题。首先，三科考试的成绩是无法比较的。由于它们的均值和标准差相差比较大，所以对它们求平均值是没有意义的。在组合这些考试成绩之前，必须将其变换为可比较的单元。其次，为了评定等级，需要一种方法来确定某个学生在前述得分上的百分比排名。再次，表示姓名的字段只有一个，这让排序任务复杂化了。为了正确地将其排序，需要将姓和名拆开。

2. 解决方案

```
> options(digits = 2)  # 保留小数位数 2 位
# 导入数据
> Student <- c("John Davis", "Angela Williams",
"Bullwinkle Moose","David Jones",
"Janice Markhammer","Cheryl Cushing",
"Reuven Ytzrhak","Greg Knox",
"Joel England","Mary Rayburn")
> Math <- c(502, 600, 412, 358, 495, 512, 410, 625, 573, 522)
> Science <- c(95, 99, 80, 82, 75, 85, 80, 95, 89, 86)
> English <- c(25, 22, 18, 15, 20, 28, 15, 30, 27, 18)
# 建立成绩表数据框
> roster <- data.frame(Student, Math, Science, English,stringsAsFactors = FALSE)
# 对学生三门课程的成绩进行标准化,赋值给 z
> z <- scale(roster[,2:4])
> z
        Math   Science   English
[1,]    0.013   1.078    0.587
[2,]    1.143   1.591    0.037
[3,]  -1.026  -0.847   -0.697
[4,]  -1.649  -0.590   -1.247
[5,]  -0.068  -1.489   -0.330
[6,]    0.128  -0.205    1.137
[7,]  -1.049  -0.847   -1.247
[8,]    1.432   1.078    1.504
[9,]    0.832   0.308    0.954
[10,]   0.243  -0.077   -0.697
```

```
attr(,"scaled:center")
  Math Science English
    501      87      22
attr(,"scaled:scale")
  Math Science English
  86.7     7.8     5.5
#计算 z 的每行平均数为每个学生的综合得分 score
> score <- apply(z, 1, mean)
#在数据库 roster 后面加上 score 变量
> roster <- cbind(roster, score)
#取分位数并赋值给 y
> y <- quantile(score, c(.8,.6,.4,.2))
> y
  80%    60%    40%    20%
0.74  0.44 -0.36 -0.89
#定义等级,分数在前 80%的等级为 A,依此类推,将等级赋值给 grade
> roster$grade[score >= y[1]] <- "A"
> roster$grade[score < y[1] & score >= y[2]] <- "B"
> roster$grade[score < y[2] & score >= y[3]] <- "C"
> roster$grade[score < y[3] & score >= y[4]] <- "D"
> roster$grade[score < y[4]] <- "F"
#把名字从空格处拆分开,即拆成姓和名
> name <- strsplit((roster$Student), " ")
> name
[[1]]
[1] "John"   "Davis"

[[2]]
[1] "Angela"   "Williams"

[[3]]
[1] "Bullwinkle" "Moose"

[[4]]
[1] "David" "Jones"

[[5]]
[1] "Janice"     "Markhammer"

[[6]]
[1] "Cheryl"   "Cushing"

[[7]]
[1] "Reuven"   "Ytzrhak"

[[8]]
[1] "Greg" "Knox"

[[9]]
[1] "Joel"     "England"
```

```
[[10]]
[1] "Mary"     "Rayburn"
#提取列表中每个成分的第 2 个元素和第 1 个元素,分别放入向量 lastname(firstname)
> lastname <- sapply(name, "[", 2)
> firstname <- sapply(name, "[", 1)
> roster <- cbind(firstname,lastname,roster[, -1])
#先按照姓再按照名进行排序
> roster <- roster[order(lastname,firstname),]
> roster
    firstname  lastname Math Science English score grade
6   Cheryl     Cushing    512    85     28     0.35   C
1   John       Davis      502    95     25     0.56   B
9   Joel       England    573    89     27     0.70   B
4   David      Jones      358    82     15    -1.16   F
8   Greg       Knox       625    95     30     1.34   A
5   Janice     Markhammer 495    75     20    -0.63   D
3   Bullwinkle Moose      412    80     18    -0.86   D
10  Mary       Rayburn    522    86     18    -0.18   C
2   Angela     Williams   600    99     22     0.92   A
7   Reuven     Ytzrhak    410    80     15    -1.05   F
```

至此,已按照要求在 R 环境下完成了问题的分析求解。

6.2 描述性统计分析

数据的描述性统计分析是通过绘制统计图形、编制统计表格、计算统计量等方法来探索数据的主要分布特征,揭示其中存在的规律。在描述性统计中,样本的观测值中含有总体各方面的信息,它来自总体,信息较为分散,显得杂乱无章。为了能够反映总体的各项特征,需要将这些分散在样本中的有关总体的信息集中起来,对样本进行加工,得到统计量。在描述性统计量的计算方面,针对不同类型的数据 R 提供了非常多的函数来获取描述性统计量。

6.2.1 描述统计函数

本节主要介绍 R 中一些常用的描述统计函数及其用法。

1. summary()

summary()提供了最小值、最大值、四分位数和数值型变量的均值,以及因子向量和逻辑型向量的频数统计,下面举例说明。

```
> summary(mtcars[vars])
      mpg              hp             wt
Min.   :10.40  Min.   : 52.0  Min.   :1.513
1st Qu.:15.43  1st Qu.: 96.5  1st Qu.:2.581
Median :19.20  Median :123.0  Median :3.325
Mean   :20.09  Mean   :146.7  Mean   :3.217
3rd Qu.:22.80  3rd Qu.:180.0  3rd Qu.:3.610
Max.   :33.90  Max.   :335.0  Max.   :5.424
```

2. sapply()

同样地,可以使用 apply()或 sapply()计算所选择的任意描述性统计量。对于 sapply(),其语法格式为:

```
sapply(x, FUN, options)
```

其中,x 是数据框(或矩阵),FUN 为一个任意的函数。如果指定了 options,它们将被传递给 FUN 。可以在这里插入典型函数,如 mean、sd、var、min、max、median、length、range 和 quantile 等。fivenum()可返回 5 数总括(即最小值、下四分位数、中位数、上四分位数和最大值)。

```
> mystats <- function(x, na.omit = FALSE){
if (na.omit)
x <- x[!is.na(x)]
m <- mean(x)
n <- length(x)
s <- sd(x)
skew <- sum((x - m)^3/s^3)/n
kurt <- sum((x - m)^4/s^4)/n - 3
return(c(n = n, mean = m, stdev = s, skew = skew, kurtosis = kurt))
}
> sapply(mtcars[vars], mystats)
            mpg       hp       wt
n         32.00    32.00   32.000
mean      20.09   146.69    3.217
stdev      6.03    68.56    0.978
skew       0.61     0.73    0.423
kurtosis  -0.37    -0.14   -0.023
```

3. aggregate()

在比较多组个体或观测时,关注的焦点经常是各组的描述性统计信息,而不是样本整体的描述性统计信息。在 R 中完成这个任务有很多方法。在 6.1.2 节中,利用 apply( )可以实现对行列的汇总,也可以使用 aggregate()来分组获取描述性统计量,代码如下所示:

```
> aggregate(mtcars[,vars], by = list(am = mtcars$am), mean)
  am  mpg  hp  wt
1  0  17  160  3.8
2  1  24  127  2.4
> aggregate(mtcars[,vars], by = list(am = mtcars$am), sd)
  am  mpg  hp  wt
1  0  3.8  54  0.78
2  1  6.2  84  0.62
```

注意:list(am=mtcars$am)的使用,如果使用的是 list(mtcars$am),那么 am 列将被标注为 Group.1 而不是 am。如果有多个分组变量,可以使用 by=list(name1=groupvar1, name2=groupvar2,…,groupvarN)语句。

遗憾的是，aggregate()仅允许在每次调用中使用平均数、标准差这样的单返回值函数，无法一次返回若干个统计量。若要实现这样的任务，可以使用 by()，其语法格式为：

```
by(data, INDICES, FUN)
```

其中，data 是一个数据框或矩阵；INDICES 是一个因子或因子组成的列表，定义了分组；FUN 是任意函数。具体如下：

```
> by(mtcars[,vars], mtcars[,"am"], summary)
mtcars[, "am"]: 0
      mpg              hp              wt
Min.   :10.40  Min.   : 62.0  Min.    :2.465
1st Qu.:14.95  1st Qu. :116.5  1st Qu. :3.438
Median:17.30  Median :175.0  Median  :3.520
Mean   :17.15  Mean   :160.3  Mean    :3.769
3rd Qu.:19.20  3rd Qu. :192.5  3rd Qu. :3.842
Max.   :24.40  Max.   :245.0  Max.    :5.424
-----------------------------------------------------------
mtcars[, "am"]: 1
      mpg              hp              wt
Min.   :15.00  Min.   : 52.0  Min.    :1.513
1st Qu.:21.00  1st Qu. : 66.0  1st Qu. :1.935
Median:22.80  Median :109.0  Median  :2.320
Mean   :24.39  Mean   :126.8  Mean    :2.411
3rd Qu.:30.40  3rd Qu. :113.0  3rd Qu. :2.780
Max.   :33.90  Max.   :335.0  Max.    :3.570
```

6.2.2 软件包的描述统计

有些用户贡献包提供计算描述性统计量的函数，其中包括 Hmisc、pastecs 和 psych。由于这些包不在安装的基础包中，故在首次使用之前先要进行安装，可通过命令 install.packages("packagename")。

1. Hmisc 包

Hmisc 包中的 describe()可返回变量和观测的数量、缺失值和唯一值的数目、平均值、分位数，以及 5 个最大的值和 5 个最小的值。下面进行举例说明。

```
> install.packages("Hmisc")
> library(Hmisc)
> describe(mtcars[vars])
mtcars[vars]
3  Variables      32  Observations
------------------------------------------------------------------------
mpg
  n missing  unique  Mean    .05    .10    .25    .50    .75    .90    .95
32       0      25  20.09  12.00  14.34  15.43  19.20  22.80  30.09  31.30

lowest : 10.4  13.3  14.3  14.7  15.0,  highest: 26.0  27.3  30.4  32.4  33.9
```

```
-----------------------------------------------------------------------
hp
  n missing  unique    Mean    .05    .10    .25    .50    .75    .90    .95
32       0      22    146.7  63.65  66.00  96.50  123.00  180.00  243.50  253.55

lowest :  52  62  65  66  91,    highest: 215  230  245  264  335
-----------------------------------------------------------------------
wt
  n missing  unique    Mean    .05    .10    .25    .50    .75    .90    .95
32       0      29    3.217  1.736  1.956  2.581  3.325  3.610  4.048  5.293

lowest : 1.513  1.615  1.835  1.935  2.140, highest: 3.845  4.070  5.250  5.345  5.424
```

2. pastecs 包

pastecs 包中的 stat.desc()可以计算种类繁多的描述性统计量，其语法格式为：

```
stat.desc(x, basic = TRUE, desc = TRUE, norm = FALSE, p = 0.95)
```

其中，x 是一个数据框或时间序列。若 basic＝TRUE(默认值)，则计算其中所有值、空值、缺失值的数量，以及最小值、最大值、值域，还有总和。若 desc＝TRUE(同样也是默认值)，则计算中位数、平均数、平均数的标准误、平均数置信度为 95％的置信区间、方差、标准差及变异系数。最后，若 norm＝TRUE(不是默认的)，则返回正态分布统计量，包括偏度和峰度(及它们的统计显著程度)，以及 Shapiro-Wilk 正态检验结果。这里使用了 p 值来计算平均数的置信区间(默认置信度为 0.95)。具体见下面的代码：

```
> install.packages("pastecs")
> library(pastecs)
> stat.desc(mtcars[vars])
                 mpg       hp      wt
nbr.val         32.0    32.00   32.00
nbr.null         0.0     0.00    0.00
nbr.na           0.0     0.00    0.00
min             10.4    52.00    1.51
max             33.9   335.00    5.42
range           23.5   283.00    3.91
sum            642.9  4694.00  102.95
median          19.2   123.00    3.33
mean            20.1   146.69    3.22
SE.mean          1.1    12.12    0.17
CI.mean.0.95     2.2    24.72    0.35
var             36.3  4700.87    0.96
std.dev          6.0    68.56    0.98
coef.var         0.3     0.47    0.30
```

3. doBy 包

在 doBy 包中 summaryBy()可进行统计性描述，其语法格式为：

```
summaryBy(formula, data = dataframe, FUN = function)
```

其中，formula 的格式为：

```
var1 + var2 + var3 + … + varN ~ groupvar1 + groupvar2 + … + groupvarN
```

在～左侧的变量是需要分析的数值型变量，而右侧是类别型的分组变量，function 可为任何内建或用户自编的 R 函数。具体的代码如下：

```
> options(digits = 2)
> install.packages("doBy")
> library(doBy)
> summaryBy(mpg + hp + wt~am, data = mtcars, FUN = mystats)
  am mpg.n mpg.mean mpg.stdev mpg.skew mpg.kurtosis
1  0    19       17       3.8    0.014         -0.8
2  1    13       24       6.2    0.053         -1.5
  hp.n hp.mean hp.stdev hp.skew hp.kurtosis wt.n
1   19     160       54  -0.014       -1.21   19
2   13     127       84   1.360        0.56   13
  wt.mean wt.stdev wt.skew wt.kurtosis
1     3.8     0.78    0.98        0.14
      2.4     0.62    0.21       -1.17
```

4. psych 包

psych 包中的 describe. by( )可计算和 describe 相同的描述性统计量，只是按照一个或多个分组变量分层，具体如下：

```
> install.packages("psych")
> library(psych)
> describe.by(mtcars[vars], mtcars$am)
group: 0
    vars  n  mean    sd median trimmed   mad  min   max range  skew kurtosis    se
mpg    1 19  17.1  3.83   17.3    17.1  3.11 10.4  24.4    14  0.01    -0.80  0.88
hp     2 19 160.3 53.91  175.0   161.1 77.10 62.0 245.0   183 -0.01    -1.21 12.37
wt     3 19   3.8  0.78    3.5     3.8  0.45  2.5   5.4     3  0.98     0.14  0.18
---------------------------------------------------------------------------------

group: 1
    vars  n  mean    sd median trimmed   mad  min   max range  skew kurtosis    se
mpg    1 13  24.4  6.17   22.8    24.4  6.67 15.0  33.9  18.9  0.05    -1.46  1.71
hp     2 13 126.8 84.06  109.0   114.7 63.75 52.0 335.0 283.0  1.36     0.56 23.31
wt     3 13   2.4  0.62    2.3     2.4  0.68  1.5   3.6   2.1  0.21    -1.17  0.17
```

6.3 多元统计分析

多元统计分析也称为多因素统计分析，是运用数理统计方法来研究解决多指标问题的理论和方法。它是研究客观事物中多变量或多因素之间的相互依赖关系及统计规律的数理统计

学分支之一，是现代统计分析理论和方法。通过采用多元统计分析技术进行数据处理、建立宏观或微观系统模型，可以实现对变量的相关性分析；构造预测模型，进行预报控制；进行数值分类，构造分类模式；简化系统结构，探讨系统内核4方面问题的解决。

多元统计分析方法包括判别分析、聚类分析、主成分分析、因子分析、对应分析、典型相关分析、多维标度法及多变量可视化分析等。其中，主成分分析与因子分析的目的是寻找多个变量的“代表”；判别分析能将对象分类到已知类别中；聚类分析按照一定的尺度把对象分类；典型相关分析研究两组变量之间的相关问题；对应分析探究行列变量的关系。

统计软件的出现使得人们能够更加简便和准确地解决实际问题，作为免费、开源的R来说，在统计方面更是有着自己独特的优势和强大的绘图功能。本章将利用R软件来探索多元统计分析方面的应用。

6.3.1　方差分析

方差分析(Analysis of Variance，ANOVA)，又称为变异数分析或F检验，是R. A. Fisher发明的，用于两个及两个以上样本均数差别的显著性检验。它是一种分析各个自变量对因变量的影响的方法，其自变量是定性变量的因子及可能出现的称为协变量的定量变量。由于各种因素的影响，研究所得的数据呈现波动状，造成波动的原因可分成两类：一类是不可控的随机因素，另一类是研究中施加的对结果造成影响的可控因素，方差分析是从观测变量的方差入手，研究诸多控制变量中哪些变量是对观测变量有显著影响的变量。首先自变量的取值不同，因变量的值也会变化，方差分析可对其进行分解，从而得出每一个自变量对结果都有一份贡献；然后把剩下的不能用已知原因解释的当作随机误差；接着对各自变量和随机误差的贡献进行F检验，从而输出F值和检验的一些p值来判断该自变量的不同水平对因变量的变化是否有显著贡献；最后会得出一个方差分析表来表示分析结果。

下面给出一个在R环境中进行方差分析的例子。

```
>x<-c(25.6,22.2,28.0,29.8,24.4,30.0,29.0,27.5,25.0,27.7,23.0,32.2,28.8,28.0,31.5,25.9,
20.6,21.2,22.0,21.2)
#数据集用5个因子水平测量,是否存在差异
#首先对数据 x 进行格式转化
>b<-data.frame(x,a=gl(5,4,20))
#得到如下结果(gl 指定因子,5 是水平,4 是重复次数)
      x a
1   25.6 1
2   22.2 1
3   28.0 1
4   29.8 1
5   24.4 2
6   30.0 2
7   29.0 2
8   27.5 2
9   25.0 3
10 27.7 3
11 23.0 3
```

```
12 32.2 3
13 28.8 4
14 28.0 4
15 31.5 4
16 25.9 4
17 20.6 5
18 21.2 5
19 22.0 5
20 21.2 5
```

在进行方差分析之前先对几条假设进行检验，由于随机抽取，假设总体满足独立、正态，考察方差齐次性（用 bartlett 检验）。

```
> bartlett.test(x~a,data=b)
#Bartlett test of homogeneity of variances
data:  x by a
Bartlett's K-squared = 7.0966, df = 4, p-value = 0.1309
```

符合方差齐次性条件，可进行方差分析。

```
> m1 <- aov(x~a,data=b)
> summary(m1)
Df Sum Sq Mean Sq F value Pr(>F)
A              4  132.0  32.99  4.306 0.0162 *
Residuals  15  114.9  7.66
Signif.codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '' 1
```

从这个结果看出差别显著。接下来考察具体的差异（多重比较）。

```
> TukeyHSD(m1)
  Tukey multiple comparisons of means
  95% family-wise confidence level
Fit: aov(formula = x ~ a, data = b)
$a
          Diff       lwr          upr        p adj
2-1     1.325   -4.718582    7.3685818  0.9584566
3-1     0.575   -5.468582    6.6185818  0.9981815
4-1     2.150   -3.893582    8.1935818  0.8046644
5-1    -5.150   -11.193582   0.8935818  0.1140537
3-2    -0.750   -6.793582    5.2935818  0.9949181
4-2     0.825   -5.218582    6.8685818  0.9926905
5-2    -6.475   -12.518582  -0.4314182  0.0330240
4-3     1.575   -4.468582    7.6185818  0.9251337
5-3    -5.725   -11.768582   0.3185818  0.0675152
5-4    -7.300   -13.343582  -1.2564182  0.0146983
```

除了 5,2 和 5,4 之间外，其他之间的差异是不显著的。

6.3.2　判别分析

判别分析又称为“分辨法”，是在分类确定的条件下，根据某一研究对象的各种特征值判别其类型归属问题的一种多变量统计分析方法。其基本原理是按照一定的判别准则建立一个或多个判别函数，用研究对象的大量资料确定判别函数中的待定系数并计算判别指标，据此即可确定某一样本属于何类。当得到一个新的样品数据，要确定该样品属于已知类型中的哪一类，这类问题属于判别分析问题。

判别分析的方法大体上有三类，即 Fisher 判别、Bayes 判别和距离判别。Fisher 判别思想是投影降维，使多维问题简化为一维问题来处理。选择一个适当的投影轴，使所有的样品点都投影到这个轴上，得到一个投影值。对这个投影轴方向的要求是：使每一组内的投影值所形成的组内离差尽可能小，而不同组间的投影值所形成的类间离差尽可能大。Bayes 判别思想是根据先验概率求出后验概率，并依据后验概率分布作出统计推断。距离判别思想是根据已知分类的数据计算各类别的重心，对未知分类的数据，计算它与各类重心的距离，与某个重心距离最近则归于该类。

1. 线性判别

当不同类样本的协方差矩阵相同时，可以在 R 中使用 MASS 包的 lda 函数实现线性判别。lda 函数以 Bayes 判别思想为基础。当分类只有两种且总体服从多元正态分布条件下，Bayes 判别与 Fisher 判别、距离判别是等价的。本例使用 iris 数据集来对花的品种进行分类。首先载入 MASS 包，建立判别模型，其中的 prior 参数表示先验概率。然后利用 table 函数建立混淆矩阵，比对真实类别和预测类别。

```
> library(MASS)
> model1 = lda(Species~.,data = iris,prior = c(1,1,1)/3)
> table(Species,predict(model1) $ class)

Species       setosa versicolor virginica
setosa          50         0         0
versicolor      0         48        2
virginica       0         1         49
```

从以上结果可观察到判断错误的样本只有三个。在判别函数建立后，还可以类似主成分分析那样对判别得分进行绘图。

```
> ld = predict(model1) $ x
> p = ggplot(cbind(iris,as.data.frame(ld)),aes(x = LD1,y = LD2))
> p + geom_point(aes(colour = Species),alpha = 0.8,size = 3)
```

2. 二次判别

当不同类样本的协方差矩阵不同时，则应该使用二次判别。

```
> model2 = qda(Species~.,data = iris,cv = T)
```

这里将 CV 参数设置为 T 是使用留一交叉检验（leave－one－out cross－validation），并自动生成预测值。这种条件下生成的混淆矩阵较为可靠。此外，还可以使用 predict (model) $ posterior 提取后验概率。

注意：在使用 lda 和 qda 函数时，其假设是总体服从多元正态分布，若不满足的话则谨慎使用。

6.3.3 聚类分析

正所谓"物以类聚，人以群分"，聚类分析是一种研究样品或变量分类问题的多元统计方法。所谓类，指的是相似元素的集合。与带有主观性和任意性的定性分类处理不同，聚类分析属于客观的利用数学方法实现的数值分类，它更加科学，能够很好地揭示客观事物内在的本质差别与联系。聚类分析是一种探索性的分析，在分类的过程中不必事先给出一个分类的标准，它能够从样本数据出发，自动对其进行分类，其结果因所使用的方法不同而不同。

在经济、管理、地质勘探、天气预报、生物分类、考古学、医学、心理学及制定国家标准和区域标准等领域或问题的研究中存在大量量化分类研究。例如在生物学中，为了研究生物的演变，生物学家需要根据各种生物不同的属性特征对生物进行分类。在人口学研究中，需要构造人口剩余分类模式，人口死亡分类情况，以此来研究人口的生育和死亡规律。

聚类分析内容非常丰富，最常用的是系统聚类法。系统聚类法的基本思想：先视 n 个样品各自为一类，然后每次将具有最小距离的两类合并成一个新类，计算合并后的类间距，重复这一过程，直到所有样品归成一类为止。

系统聚类方法分为最短距离法、最长距离法、中间距离法、重心法、类平均法、可变类平均法、可变法、离差平方和法（Ward 法）。

R 软件及其相关包提供了各种聚类方法，其中系统聚类的程序语句如下：

```
hclust(d, method = "complete", members = NULL)
```

其中，d 为距离计算方法，包括绝对值距离、欧氏距离、切比雪夫距离、马氏距离、兰氏距离等，默认为欧氏距离；method 包括 ward（离差平方和法）、single（最短距离法）、complete（最长距离法）、average（类平均法）、median（中间距离法）及 centroid（重心法）。

由于 R 语言的系统聚类函数选项较多，现编制一个简便的函数进行快速聚类分析。程序语句如下：

```
H.clust <- function(x, d = "euclidean", method = "complete", process = F, plot = T)
```

其中，x 是数据框或数值矩阵；d 为距离计算方法（同上）；method 为系统聚类方法（同上）；process 为是否输出聚类过程；plot 为是否输出树状图。

下面是一个聚类分析实证，利用 R 通过系统聚类法对 2012 年全国 31 个省、市、自治区的居民收入与消费水平进行分类。样本数据来自国泰安 CSMAR 数据库，选取 2012 年居民收入与消费数据为研究对象，包括农村居民消费水平、城镇居民消费水平、城镇居民家庭平均每人全年总收入、农村居民家庭平均每人全年纯收入、城镇居民消费支出，农村居民消费支出、城乡储蓄数据。数据的具体下载步骤见附录一。

R 语言程序如下：

```
#导入数据
> install.packages('xlsx')
> library(xlsx)
> juldata <- read.xlsx("C:\\Users\\min.li\\Documents\\data.xlsx",1,header = T)
> x <- juldata[,c(4:10)]
#标记列名
> row.names(x)<-c("北京","天津","河北","山西","内蒙古","辽宁","吉林",
"黑龙江","上海","江苏","浙江","安徽","福建","江西","山东","河南","湖北","湖南","广东","
广西","海南","重庆","四川","贵州","云南","西藏","陕西","甘肃","青海","宁夏","新疆")
#计算标准化后的距离
> d <- dist(scale(x))
#离差平方和法
> y <- hclust(d,"ward")
> plot(y,hang = -1)
```

得到的聚类图如图 6.1 所示。

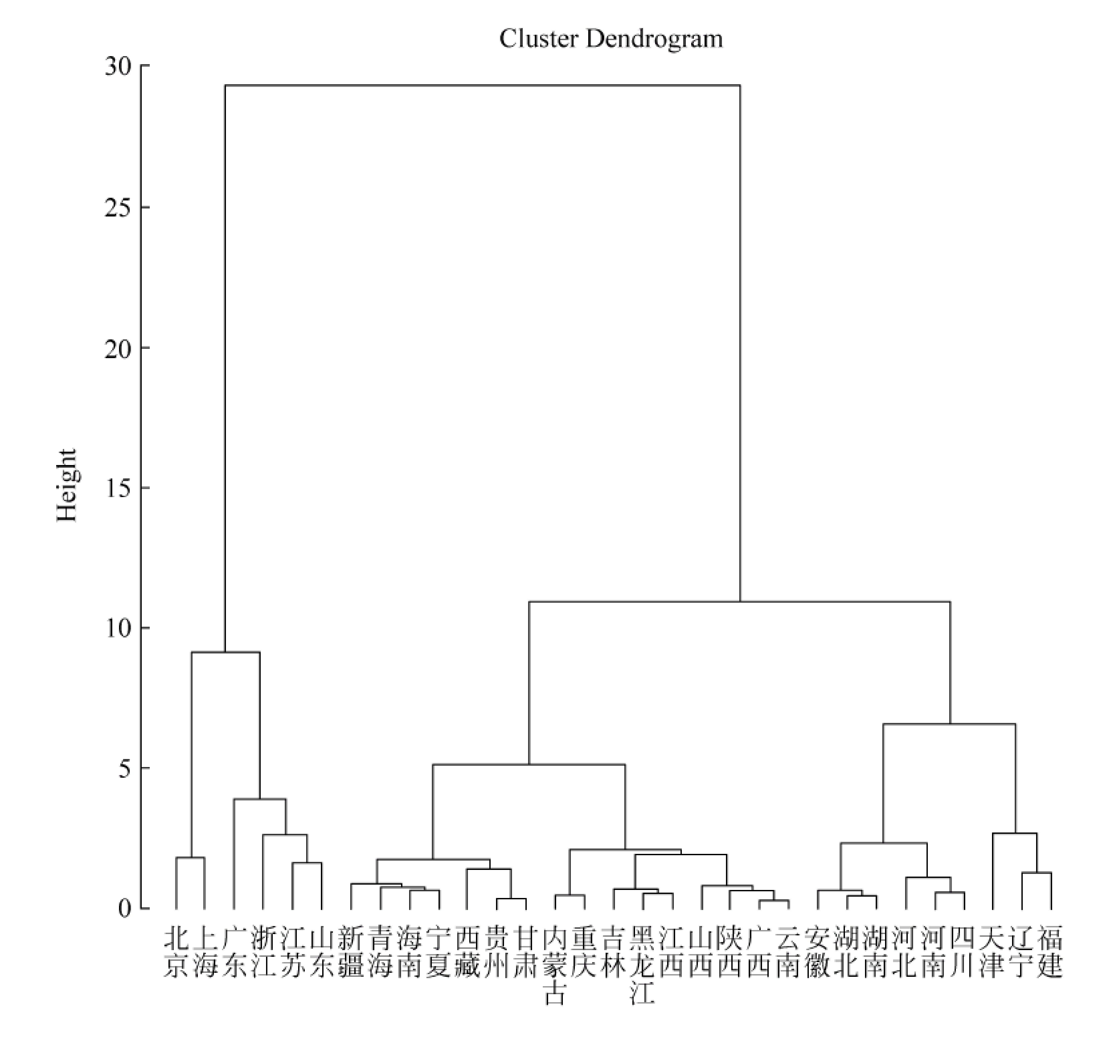


图 6.1　离差平方和法的聚类图

根据离差平方和法聚类分析得到的树状图,可以得到如下几种分类。从我国各省、市、自治区的消费情况来看,分为三类比较合适,如表 6.5 所示。

表 6.5 按类整理聚类结果

<table>
<tr><th>分类</th><th colspan="2">第 一 类</th><th colspan="2">第 二 类</th><th colspan="2">第 三 类</th></tr>
<tr><td>分三类</td><td colspan="2">北京 上海
广东 浙江
江苏 山东</td><td colspan="2">新疆 青海 海南 宁夏 西藏
贵州 甘肃 内蒙古 重庆 吉林
黑龙江 江西 山西 陕西
广西 云南</td><td colspan="2">安徽 湖北 湖南 河北 河南
四川 天津 辽宁 福建</td></tr>
<tr><td rowspan="2">分四类</td><td colspan="2">第一类</td><td colspan="2">第二类</td><td>第三类</td><td>第四类</td></tr>
<tr><td colspan="2">北京 上海</td><td colspan="2">广东 浙江
江苏 山东</td><td>新疆 青海 海南
宁夏 西藏 贵州
甘肃 内蒙古 重
庆 吉林 黑龙江
江西 山西 陕西
广西 云南</td><td>安徽 湖北 湖南
河北 河南 四川
天津 辽宁 福建</td></tr>
<tr><td rowspan="2">分六类</td><td>第一类</td><td>第二类</td><td>第三类</td><td>第四类</td><td>第五类</td><td>第六类</td></tr>
<tr><td>北京
上海</td><td>广东 浙江
江苏 山东</td><td>新疆 青海
海南 宁夏
西藏 贵州
甘肃</td><td>内蒙古 重庆
吉林 黑龙江
江西 山西
陕西 广西
云南</td><td>安徽 湖北 湖南
河北 河南 四川</td><td>天津 辽宁 福建</td></tr>
</table>

从表 6.5 可以看出，北京、上海、广东、浙江、江苏和山东等地的居民收入和消费水平要明显优于其他省、市、自治区，而西部地区收入和消费水平比较低，这和经济水平是相一致的。较高的经济水平可以提高居民的收入和消费水平，所以必须注重经济的发展。由以上实证分析可以看出，利用 R 软件可以简单方便地实现系统聚类分析，而且还根据不同的情况编写程序，实现一些特殊的分析功能。

6.3.4 主成分分析

主成分分析也称为主分量分析，是指通过数据分析寻求使用较少的变量去解释原来数据中的大部分变异的一种统计分析方法。它是由 Person(1901)提出，后来被 Hotelling(1933)发展起来的。主成分分析就是从事物错综复杂的关系中找出部分主成分进行定量分析，从而实现降维和简化的作用。

R 语言主成分分析的程序语句如下：

```
princomp(x,cor = FALSE,scores = TRUE, … )  #主成分分析函数
```

其中，x 为数据矩阵；cor 为是否用相关阵，默认为协差阵；scores 为是否输出成分得分。

```
screeplot(obj,type = c("barplot","lines"), … )  #碎石土函数
```

其中，obj 为主成分分析对象；type 为图形类型。

下面利用 R 通过主成分分析方法对 6.3.2 节中来自 CSMAR 数据库的 2012 年全国 31 个省、市、自治区的居民收入与消费水平进行评分。

R 语言程序如下：

```
#主成分分析函数
> pca = princomp(scale(x),cor = T)
> summary(pca,loadings = T)
Importance of components:
                        Comp.1     Comp.2     Comp.3      Comp.4       Comp.5
Standard deviation      2.2080698  1.3315395  0.45193423  0.258239995  0.18571101
Proportion of Variance  0.6965103  0.2532853  0.02917779  0.009526842  0.00492694
Cumulative Proportion   0.6965103  0.9497956  0.97897343  0.988500272  0.99342721
                        Comp.6       Comp.7
Standard deviation      0.158231094  0.144818639
Proportion of Variance  0.003576726  0.002996063
Cumulative Proportion   0.997003937  1.000000000
Loadings:
          Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7
Inco1002  -0.409  -0.286  -0.345           0.468   0.330   0.549
Inco1003  -0.418  -0.241   0.280  -0.477   0.394  -0.337  -0.438
Inco0202  -0.425  -0.225          -0.342  -0.752   0.293
Inco0301  -0.411  -0.262  -0.329   0.656  -0.151  -0.387  -0.227
Inco0904  -0.366   0.408   0.445  -0.418   0.559
Inco0903  -0.190   0.643  -0.640  -0.317  -0.138  -0.143
Inco0101  -0.373   0.402   0.293   0.339   0.168   0.590  -0.351
```

由结果可以看出，第一个主成分的方差贡献率为 69.65%，第二个主成分的方差贡献率为 25.33%，前两个主成分的累计方差贡献率为 94.98%，另外的 5 个主成分可以舍去。

```
#画出碎石图(如图 6.2 所示)
> screeplot(pca,type = "line",main = "碎石图")
```

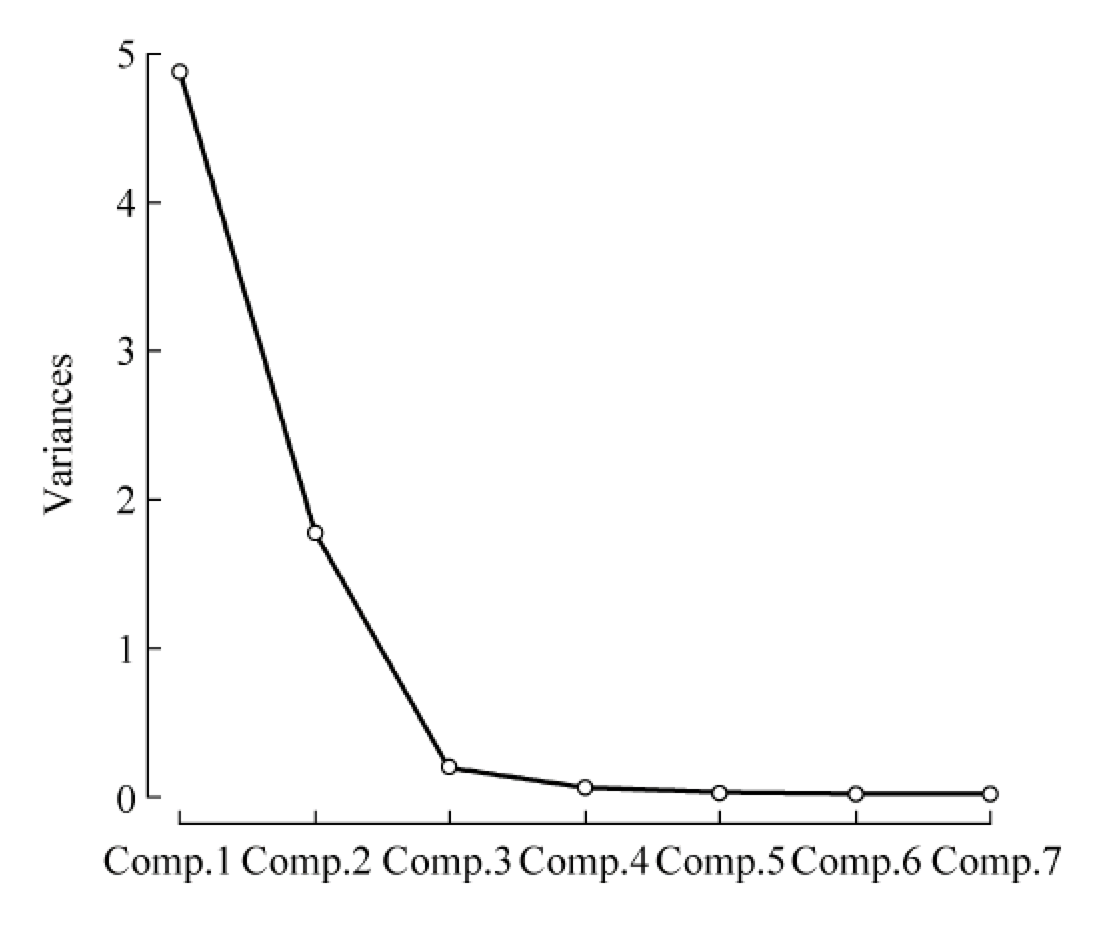


图 6.2　碎石图

碎石图是一种可以帮助确定主成分合适个数的有用的视觉工具，将特征值从大到小排列。从碎石图中同样可以看出，前两个主成分的累积方差贡献率已经占了较大比重，所以选取两个主成分。

```
#计算前两个主成分得分
> s1 = pca $ scores[1:31,1]
> s2 = pca $ scores[1:31,2]
#计算综合得分
> c = ( 0.6965 * s1 + 0.2533 * s2)/( 0.6965 + 0.2533)
#排序
> r = rank(c)
> cbind(s1,s2,c,r)
```

通过提取前两个主成分，计算各个省、市、自治区的得分，可以给出它们科技发展水平的一个排名。从表中结果可以看出，上海第一，北京第二，浙江第三，广东、江苏和天津紧随其后，都属于第一梯队。这些地区都是经济发展水平排在全国前列的。排在最后的基本上都是经济相对落后的西部地区，科技发展水平也比其他地区要低。主成分分析得分及排名如表 6.6 所示。

表 6.6 主成分分析得分及排名

地区	s1	s2	c	r
北京	−4.088524	−2.506376	−3.666585	2
天津	−1.541813	−2.449816	−1.783967	6
河北	0.2235620	1.3305248	0.5187754	15
山西	1.2281585	0.1494170	0.9404714	22
内蒙古	0.6320788	−0.792218	0.2522362	10
辽宁	−0.847345	−0.009200	−0.623822	9
吉林	1.0537860	−0.530619	0.6312445	17
黑龙江	1.1592341	−0.052002	0.8362123	19
上海	−5.575225	−3.136581	−4.924868	1
江苏	−3.428374	1.5847985	−2.091422	5
浙江	−4.163469	0.0703022	−3.034374	3
安徽	0.8075467	0.5575598	0.7408783	18
福建	−0.753069	−0.629552	−0.720129	8
江西	1.1194688	0.1962514	0.8732581	21
山东	−1.983709	2.3148333	−0.837340	7
河南	0.1446742	1.8509641	0.5997209	16
湖北	0.3970400	0.6153339	0.4552563	11
湖南	0.3695942	0.9125549	0.5143952	14
广东	−4.530909	2.9720216	−2.529970	4
广西	1.2094716	0.2109302	0.9431729	23
海南	1.8359584	−1.078039	1.0588310	24
重庆	0.8362220	−0.510601	0.4770407	12
四川	−0.006544	1.8651337	0.4926088	13
贵州	2.1699437	0.0307301	1.5994417	29

续表

地区	s1	s2	c	r
云南	1.3515005	0.3334030	1.0799865	26
西藏	2.8858787	−0.758998	1.9138348	31
陕西	1.1290203	0.0904629	0.8520499	20
甘肃	2.3947925	−0.087834	1.7327064	30
青海	2.3542706	−0.927091	1.4791715	28
宁夏	1.8669094	−1.147052	1.0631228	25
新疆	1.7498761	−0.469234	1.1580665	27

主成分分析方法是一种能够降低和减少各指标之间的信息冗余，简化问题的结构，提高问题分析的效率定量分析方法。借助 R 软件可以方便快捷地实现对实际问题的主成分分析计算。

6.3.5　因子分析

因子分析是从 Charles Spearman 在 1904 年发表的文章《对智力测验得分进行统计分析》开始，他提出这种方法用来解决智力测验得分的统计方法。目前因子分析在心理学、社会学、经济学等学科中都取得了成功的应用，是多元统计分析中的典型方法之一。

因子分析和主成分分析一样是一种降维、简化数据的技术。它从研究变量内部相关的依赖关系出发，探求观测数据中的基本结构，并用少数几个“抽象”的变量来表示其基本的数据结构。这几个抽象的变量被称作“因子”，能反映原来众多变量的主要信息。因子分析主要用来描述隐藏在一组可以测量到的变量中的一些更根本的，但又无法直接测量到的隐性变量。

因子分析的数学模型：

$$X_i = a_{i1}F_1 + a_{i2}F_2 + \cdots + a_{im}F_m + \varepsilon_i \quad (i = 1,2,\cdots,p)$$

其中 $F_1,F_2,\cdots,F_m$ 称为公共因子，ε_i 称为 X_i 的特殊因子，该模型可用矩阵表示为：

$$\boldsymbol{X} = \boldsymbol{AF} + \varepsilon$$

其中

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pm} \end{bmatrix} = (A_1, A_2, \cdots, A_m)$$

$$\boldsymbol{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{bmatrix}, \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_p \end{bmatrix}$$

模型中的 a_{ij} 称为因子“载荷”，是第 i 个变量在第 j 个因子上的负荷，如果把变量 X_i 看成 m 维空间中的一个点，那么 a_{ij} 表示它在坐标轴 F_j 上的投影，因此矩阵 A 称为因子载荷矩阵。

R 语言对矩阵进行极大似然因子分析的程序语句如下：

```
factanal(x,factors,scores = c("none","regression","Bartlett"),rotation = "varimax", … )
```

其中，x 为数值矩阵；factors 为因子个数；scores 为因子得分的计算方法，包括 regression 和 Bartlett 方法；rotation 为因子旋转方法。

下面利用 R 通过因子分析方法对 6.3.4 节中来自 CSMAR 数据库的 2012 年全国 31 个省、市、自治区的居民收入与消费水平进行评分。

R 语言程序如下：

```
#计算相关系数矩阵
>cor(x)
          Inco1002   Inco1003   Inco0202   Inco0301   Inco0904   Inco0903
Inco1002  1.0000000  0.93610636 0.9508626  0.9634380  0.4925974  0.09531430
Inco1003  0.9361064  1.00000000 0.9623606  0.9131647  0.5922177  0.08743624
Inco0202  0.9508626  0.96236056 1.0000000  0.9400107  0.5958519  0.14100959
Inco0301  0.9634380  0.91316472 0.9400107  1.0000000  0.5190463  0.11221110
Inco0904  0.4925974  0.59221768 0.5958519  0.5190463  1.0000000  0.74322330
Inco0903  0.0953143  0.08743624 0.1410096  0.1122111  0.7432233  1.00000000
Inco0101  0.5203417  0.59438626 0.6054109  0.5493496  0.9727392  0.75584921
          Inco0101
Inco1002  0.5203417
Inco1003  0.5943863
Inco0202  0.6054109
Inco0301  0.5493496
Inco0904  0.9727392
Inco0903  0.7558492
Inco0101  1.0000000
```

由结果可以看到变量间的相关性较强，可以通过因子分析降维、简化数据。它从研究变量内部相关的依赖关系出发，探求观测数据中的基本结构，并用少数几个“抽象”的变量来表示其基本的数据结构。这几个抽象的变量被称作“因子”，能反映原来众多变量的主要信息。先提取两个因子，代码及结果如下：

```
#极大似然法进行因子分析
>FA0 = factanal(x,2,rot = "none")
Call:
factanal(x = x,factors = 2, rotation = "none")
Uniquenesses:
Inco1002 Inco1003 Inco0202 Inco0301 Inco0904 Inco0903 Inco0101
  0.042    0.063    0.036     0.068    0.038     0.261    0.015
Loadings:
         Factor1  Factor2
Inco1002  0.809    0.551
Inco1003  0.853    0.457
Inco0202  0.867    0.461
Inco0301  0.820    0.509
Inco0904  0.894   -0.403
Inco0903  0.533   -0.675
```

```
Inco0101   0.910   -0.395
               Factor1 Factor2
SS loadings      4.720   1.757
Proportion Var   0.674   0.251
Cumulative Var   0.674   0.925
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 21.59 on 8 degrees of freedom.
The p-value is 0.00573
```

由结果可以看出,前两个因子累计方差贡献率为 86.5%。p-value 也很显著,基本上能反映所有指标信息,但各个因子的现实意义并不明显,对因子进行旋转。

```
#因子旋转
>FAC=factanal(x,2,rot="varimax")
Call:
factanal(x = x, factors = 2, rotation = "varimax")
Uniquenesses:
Inco1002 Inco1003 Inco0202 Inco0301 Inco0904 Inco0903 Inco0101
  0.042    0.063    0.036    0.068    0.038    0.261    0.015
Loadings:
         Factor1 Factor2
Inco1002   0.970   0.132
Inco1003   0.940   0.232
Inco0202   0.953   0.238
Inco0301   0.950   0.171
Inco0904   0.395   0.898
Inco0903           0.858
Inco0101   0.412   0.903
               Factor1 Factor2
SS loadings      3.963   2.514
Proportion Var   0.566   0.359
Cumulative Var   0.566   0.925
Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 21.59 on 8 degrees of freedom.
The p-value is 0.00573
```

从上述结果可以看出,旋转后的因子载荷发生了改变,各因子代表的实际意义也变得明显,如表 6.7 所示。

表 6.7　旋转后的因子载荷

变量名	公 共 因 子	
	Factor1	Factor2
Inco1002	0.970	0.132
Inco1003	0.940	0.232
Inco0202	0.953	0.238
Inco0301	0.950	0.171

续表

变量名	公共因子	
	Factor1	Factor2
Inco0904	0.395	0.898
Inco0903		0.858
Inco0101	0.412	0.903

因子 F1 代表变量 Inco1002、Inco1003、Inco0202 和 Inco0301，因子 F2 代表变量 Inco0904、Inco0903 和 Inco0101。下面计算因子得分：

```
#计算因子得分
> Fac = factanal(x,2,scores = "regression")
```

然后根据累计方差贡献率计算出每个城市的得分：

```
#计算每个地区得分
> f1 = Fac $ scores[,1]
> f2 = Fac $ scores[,2]
> F = (0.566 * f1 + 0.359 * f2)/0.925
#排序
> r2 = rank( - F)
> cbind(f1,f2,F,r2)
```

因子分析得分及排名如表 6.8 所示。

表 6.8　因子分析得分及排名

地区	f1	f2	f3	r
北京	2.4165348	−0.319547	1.3546391	3
天津	1.5344477	−1.281083	0.4417171	7
河北	−0.656002	0.9545618	−0.030929	11
山西	−0.531985	−0.010777	−0.329700	19
内蒙古	0.0427425	−0.644991	−0.224172	15
辽宁	0.2172215	0.4045423	0.2899222	8
吉林	−0.227706	−0.560420	−0.356835	20
黑龙江	−0.503889	−0.194656	−0.383873	21
上海	3.2983506	−0.675108	1.7562189	1
江苏	0.7849045	1.5744445	1.0913313	5
浙江	1.6774332	0.7359554	1.3120380	4
安徽	−0.493629	0.1023413	−0.262328	16
福建	0.6606102	−0.470533	0.2216042	9
江西	−0.465407	−0.257150	−0.384581	22
山东	−0.021432	1.5742125	0.5978505	6
河南	−0.687195	0.8476884	−0.091494	12
湖北	−0.377250	0.2481883	−0.134512	13
湖南	−0.421096	0.2433530	−0.163218	14

续表

地区	f1	f2	f3	r
广东	0.3081096	3.5482471	1.5656332	2
广西	−0.491941	−0.256308	−0.400490	23
海南	−0.287809	−1.042587	−0.580744	26
重庆	−0.179616	−0.394840	−0.263146	17
四川	−0.572151	0.8850139	−0.006613	10
贵州	−0.824072	−0.487953	−0.693621	28
云南	−0.537626	−0.284571	−0.439413	24
西藏	−0.865609	−0.973285	−0.907399	31
陕西	−0.493812	−0.066407	−0.327933	18
甘肃	−0.912643	−0.482074	−0.745536	30
青海	−0.549940	−1.029716	−0.736145	29
宁夏	−0.297848	−1.080554	−0.601622	27
新疆	−0.541687	−0.605981	−0.566640	25

通过提取因子，计算各个省、市、自治区的得分，也得到它们居民收入与消费水平的一个排名(见表6.8)。可以看到结果与主成分分析的结果是一致的，也进一步验证了结果的正确性。

因子分析方法是一种定量的测量方法，能够降低和减少各指标之间的信息冗余，简化问题的结构，提高问题分析的效率，同时也是一种可行有效的综合评价方法。借助R软件可以方便快捷地用因子分析方法对实际问题进行分析计算。

6.3.6　典型相关分析

典型相关分析(Canonical Correlation Analysis)就是利用综合变量对之间的相关关系来反映两组指标之间整体相关性的多元统计分析方法。它的基本原理是为了从总体上把握两组指标之间的相关关系，分别在两组变量中提取有代表性的两个综合变量U_1和V_1(分别为两个变量组中各变量的线性组合)，利用这两个综合变量之间的相关关系来反映两组指标之间的整体相关性。

数学描述：

考虑两组变量的向量$\boldsymbol{Z}=(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_p,\boldsymbol{Y}_1,\boldsymbol{Y}_2,\cdots,\boldsymbol{Y}_p)$，其协方差阵为：

$$\sum=\begin{bmatrix}\sum 11 & \sum 12\\ \sum 21 & \sum 22\end{bmatrix}\begin{matrix}p\\ q\end{matrix}$$
$$\quad p \qquad q$$

其中$\sum 11$是第一组变量的协方差矩阵；$\sum 22$是第二组变量的协方差矩阵；$\sum 12=\sum 21$是X和Y的协方差矩阵。那么两组变量的第一对线性组合为：

$$u_1=a'_1X \quad v_1=b'_1Y$$

其中

$$a_1=(a_{11},a_{21},\cdots,a_{p1})'$$

$$b_1 = (b_{11}, b_{21}, \cdots, b_{p1}')$$

$$\mathrm{Var}(U_1) = a_1' \mathrm{Var}(X) a_1 = a_1' \sum{}_{11} a_1 = 1$$

$$\mathrm{Var}(V_1) = b_1' \mathrm{Var}(Y) b_1 = b_1' \sum{}_{22} b_1 = 1$$

$$\rho_{U_1,V_1} = \mathrm{Cov}(U_1, V_1) = a'_1 \mathrm{Cov}(X, Y) b_1 = a_1' \sum{}_{12} b_1$$

所以典型相关分析就是求 a_1 和 b_1，使得 ρ_{uv} 达到最大。

R 语言中进行典型相关分析的函数如下：

```
cancor(x,y,xcenter = TRUE,ycenter = TRUE)
```

其中，参数 X 和 Y 表示进行相关分析的数据，为向量或矩阵形式；xcenter 和 ycenter 为逻辑变量，若值设置为 TRUE，表示将数据中心化。

下面是一个在 R 环境下进行相关分析的例子，数据来自国泰安 CSMAR 数据库，选取宏观经济指标（年度）下拉列表中的国内生产总值、工业总产值数据，时间选择 1996～2012 年。

R 语言程序如下：

```
#导入数据
>xgfxdata <- read.xlsx("C:\\Users\\min.li\\Documents\\xgfxdata.xlsx",1,header = T)
>data <- juldata[,c(2,3)]
>x <- data[,1]
>y <- data[,2]
>co <- cancor(x,y)        #典型相关分析
$cor                      #典型相关系数
[1] 0.998242
$xcoef                    #对应于 x 的系数或关于 x 的典型载荷
             [,1]
[1,] 2.041814e-06
$ycoef
            [,1]
[1,] 1.03102e-06
$xcenter                  #x 的样本均值
[1] 196610
$ycenter
[1] 279964.6
# 计算数据在典型变量下的得分
>U <- as.matrix(x) %*% co$xcoef
>V <- as.matrix(y) %*% co$ycoef
#画出变量散点图
plot(U,V,xlab = "U",ylab = "V")
```

由图 6.3 可以看出,典型相关变量呈线性相关关系,表明国内生产总值和工业总产值呈较强的线性相关性。

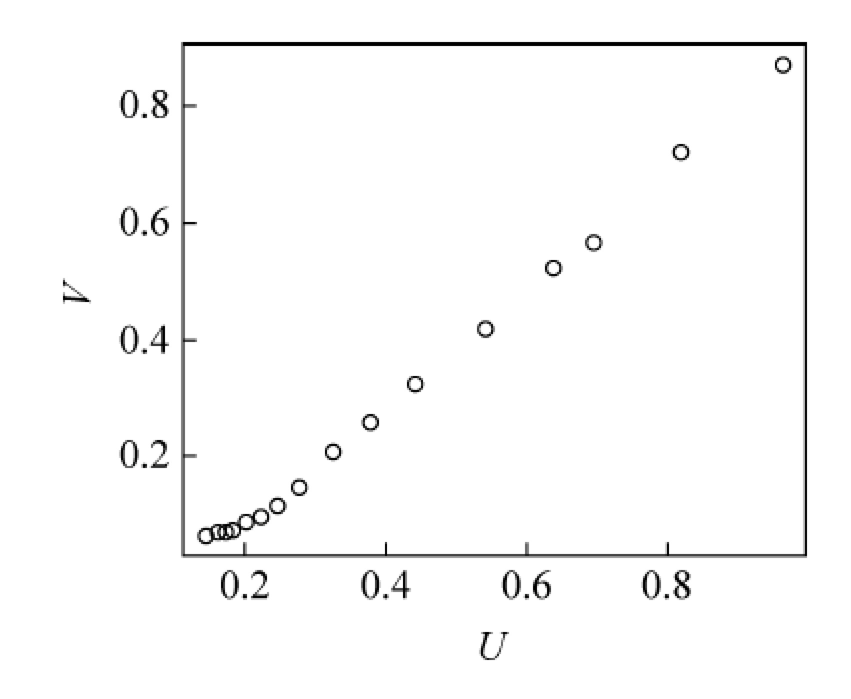


图 6.3　典型相关变量散点图

PART

3

第三部分

专题实证研究

CHAPTER 7

第7章 金融时间序列建模专题

由于 R 具备很多金融计量研究领域的包,使得没有编程经验的金融从业人员进行数据分析工作变得更为简单。本章将以具体的案例作为线索,利用 R 语言来实现整个金融数据的分析过程。

7.1 金融时间序列

时间序列指的是指标按照时间顺序将对应的数值排列的数列。时间序列分析理论是由 Andei Kolmogonor 在 1930 年提出的,随着人们对时间序列问题的研究,使得该理论得到发展和完善。时间序列分析是一种动态的统计方法,该方法的大概思想是根据已有的历史数据预测未来时间点上的数值,进而得到时间序列变化的规律。而金融时间序列指的是在某个时期内按照时间顺序进行排列的金融随机变量,主要研究资产价值随时间变化的相关理论及应用。金融时间序列表现出较强的非线性、自相关性、异方差性及随机性等。

对时间序列建立模型之前需要进行时间序列预处理,即对序列进行平稳性检验及纯随机性检验。

平稳性是时间序列分析的基础,是进行时间序列分析得到准确预测结果的前提。平稳性确保了时间序列的结构不随时间的变化而改变。如果时间序列$\{y_t\}$满足以下条件,那么该时间序列是平稳的。

(1) 时间序列的均值函数$E(y_t)=\mu$,其中 μ 为常数(与 t 无关);

(2) 时间序列的方差函数$D(y_t)=\sigma_y^2$,其中 σ_y^2 为常数(与 t 无关);

(3) 时间序列的协方差 $\mathrm{cov}(y_t, y_{t+k})=\gamma_k$,其中 γ_k 为与时间间隔 k 相关而与时间 t 无关的常数。

通常平稳性检验采用以下两种方法:

(1) 图检验方法,即根据时序图及自相关图进行检验。

时序图指的是一个以时间为横坐标,与时间对应的数值为纵坐标的二维图。如果时间

序列平稳，那么时序图表现为在某一个常数水平有相似的幅度波动且波动的范围是有界的。而自相关图是一个平面二维坐标悬垂线图。如果时间序列平稳，那么由于平稳序列一般有短期的自相关性，自相关系数会随着延迟期数的增加以较快的速度衰减到0，反之自相关系数衰减到0的速度会比较慢。

（2）统计检验方法，即构造检验统计量进行检验。

一般采用的统计检验方法是单位根检验，该方法检验时间序列平稳性的准则是判断特征根是否在单位圆内，若特征根在单位圆内，则为平稳序列。

当时间序列通过平稳性检验，对非平稳性序列进行平稳化处理后便可以建模，进行时间序列分析。

下面运用R进行时间序列的平稳性检验，样本数据来自国泰安CSMAR数据库，选取2005-1-4—2005-12-30上证综合指数(000001)的收盘价格指数为研究对象。

```
 # 0. 初始化
> setwd('E:/R-modeling/Chapter07/data')
 # 1. 读取数据
> library(RODBC)                                          # 加载包
> WH_data <- odbcConnectExcel("000001.xls")
> WH.CZCE <- sqlFetch(WH_data, "000001data")              # 读取 Excel 数据
> Data.Clpr <- data.frame(WH.CZCE[,2],WH.CZCE[,6])        # 日期,收盘价
> head(Data.Clpr)
          Date       Clpr
1   2005-01-04   1242.774
2   2005-01-05   1251.937
3   2005-01-06   1239.430
4   2005-01-07   1244.746
5   2005-01-10   1252.401
6   2005-01-11   1257.462
> tail(Data.Clpr)
          Date         Clpr
237   2005-12-23   1144.871
238   2005-12-26   1156.823
239   2005-12-27   1154.288
240   2005-12-28   1157.034
241   2005-12-29   1169.862
242   2005-12-30   1161.057
> plot.ts(Date,Clpr,type="l")                             # 收盘价时间序列图
```

由图7.1可以看出，收盘价的波动幅度较大，显示非平稳特性。图检验方法是一种简便的平稳性检验方法，其缺点是具有一定的主观性，因而得到的判别结果不能确保正确性，所以通常采用单位根检验的方法进行平稳性的判定。

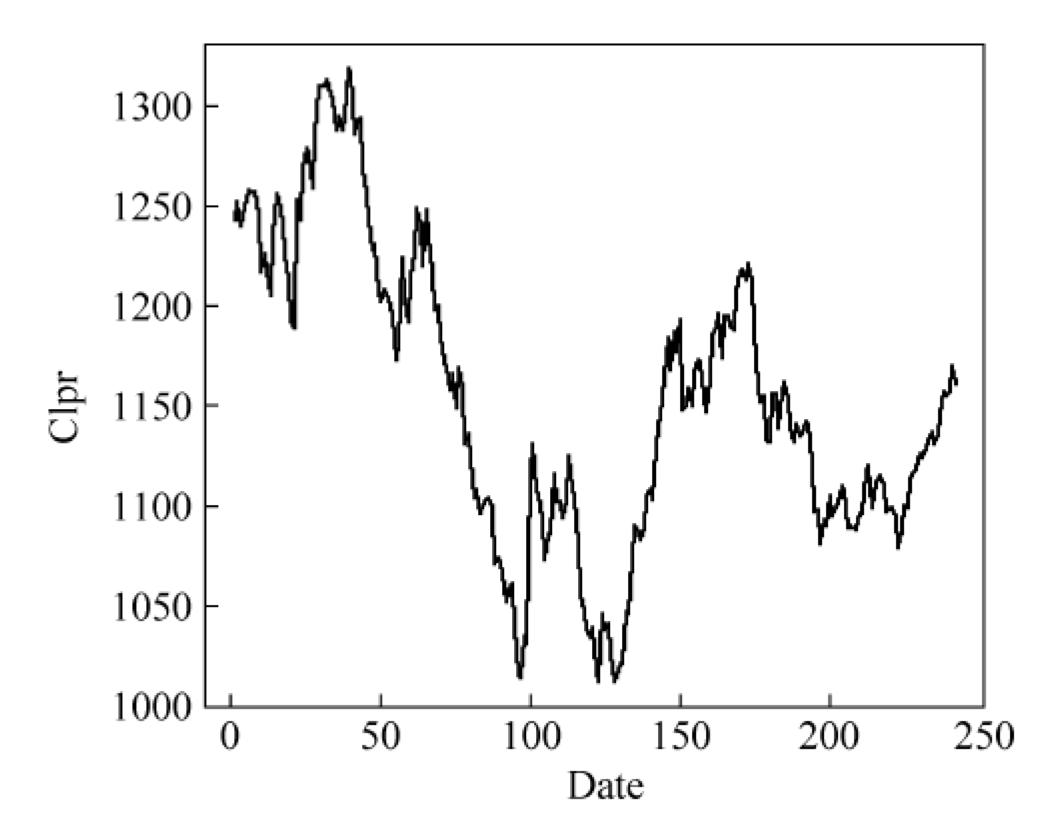


图 7.1　收盘价时间序列图

```
>library(tseries)
>adf.test(Clpr)      # 收盘价序列平稳性检验,ADF 检验
        Augmented Dickey-Fuller Test
data: Clpr
Dickey-Fuller = -1.5878, Lag order = 6, p-value = 0.7493
alternative hypothesis: stationary
```

单位根检验表明,在5%置信水平下,p-value = 0.7493>0.05,接受原假设,表明收盘价指数序列存在非平稳性。所以需要对时间序列进行处理以消除非平稳性,一般的方法是对原始数据先取对数,然后进行差分,从而得到具有稳定性的对数收益率时间序列(见图 7.2)。

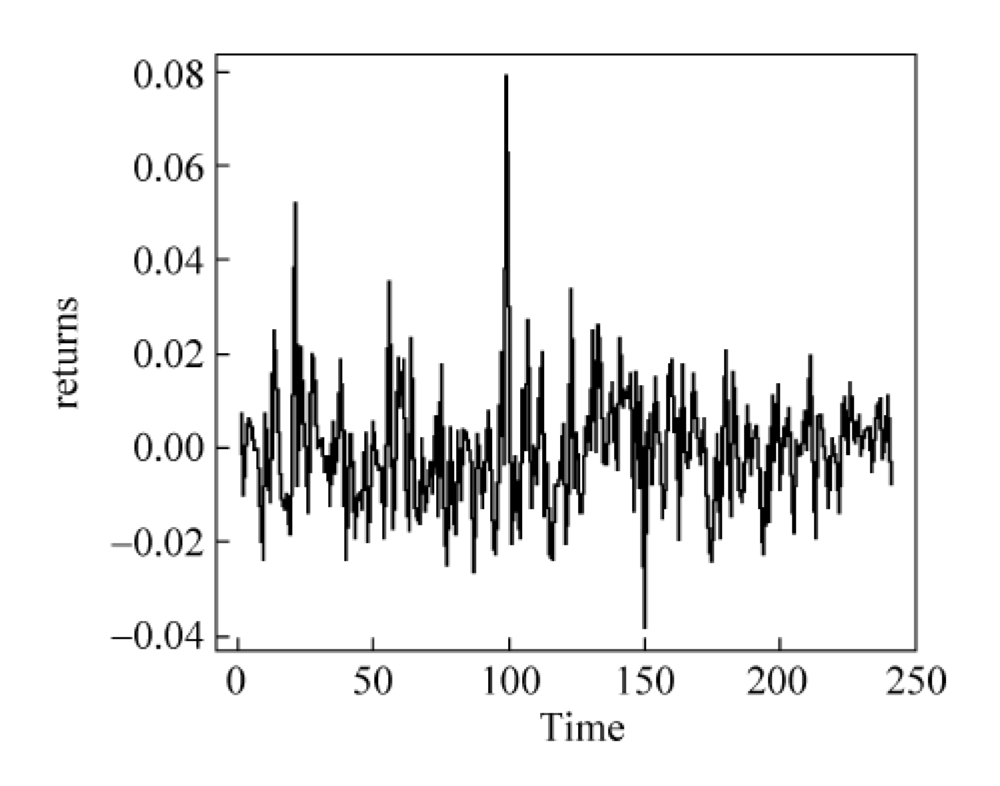


图 7.2　对数收益率时间序列图

```
>adf.test(returns)
Augmented Dickey-Fuller Test
data:  returns
Dickey-Fuller = -5.9975, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary
```

单位根检验表明，在5%显著性水平下，p－value ＝ 0.01＜0.05，拒绝原假设，表明对数收益率时间序列平稳。

7.2 ARMA 模型

ARMA 模型(Auto-Regressive and Moving Average Model)是研究时间序列的重要方法，也是目前最常用的时间序列分析模型。ARMA 模型是以自回归模型(AR 模型)和滑动平均模型(MA 模型)为基础“混合”而成的。

7.2.1 ARMA 模型简介

ARMA 模型具有以下三种基本形式：

1. AR 模型

自回归模型 AR(p)：如果时间序列 y_t 满足

$$y_t = \beta_1 y_{t-1} + \cdots + \beta_p y_{t-p} + \varepsilon_t$$

其中 ε_t 是独立同分布的随机变量序列，且满足：

$$E(\varepsilon_t) = 0, \quad \mathrm{Var}(\varepsilon_t) = \sigma_\varepsilon^2 > 0$$

则称时间序列为 y_t 服从 p 阶的自回归模型。

2. MA 模型

移动平均模型 MA(q)：如果时间序列 y_t 满足

$$y_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \cdots - \theta_q \varepsilon_{t-q}$$

则称时间序列 y_t 服从 q 阶移动平均模型。

3. ARMA 混合模型

混合模型 ARMA(p,q)：如果时间序列 y_t 满足：

$$y_t = \theta_1 y_{t-1} + \cdots + \theta_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_q \varepsilon_{t-q}$$

则称时间序列为 y_t 服从(p,q)阶自回归滑动平均混合模型。

特殊情况：$q=0$，模型即为 AR(p)，$p=0$，模型即为 MA(q)。

7.2.2 ARMA 模型定阶

自相关函数的截尾阶数确定 MA 的阶数 q，偏自相关函数的截尾阶数确定 AR 的阶数 p，因此可以用自相关和偏自相关图确定 ARMA(p,q)的阶数。

1. 自相关

如果样本的自相关系数(ACF)在滞后 $q+1$ 阶处突然截断，即在 q 处截尾，那么可以认为该序列为 MA(q)序列(见图 7.3)。

2. 偏自相关

如果样本的偏自相关系数(PACF)在滞后 p 处截尾，那么可以判定该序列为 AR(p)序列(图 7.4)。

接着 7.1.1 节对平稳性的收益率时间序列进行建模分析。

```
> acf(returns)
> pacf(returns, lag = 50)
```

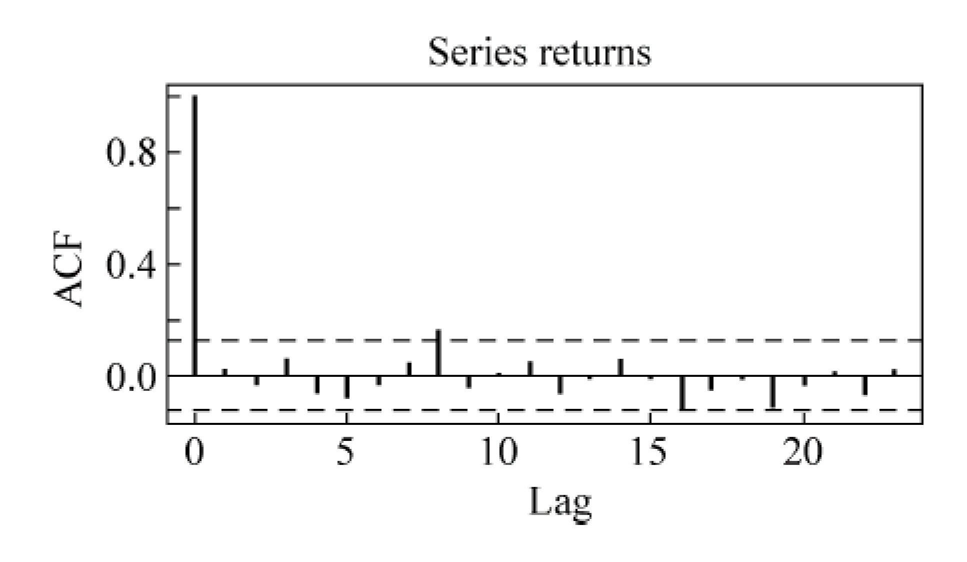


图 7.3　自相关函数图

图 7.3 自相关函数图的横轴 lag 表示滞后阶数，纵轴表示对应各阶的相关系数，0 阶滞后表示对自己的自相关系数，所以一般对应的相关系数值为 1。图中上下的虚线内为 95% 置信区间，若 lag>0 对应的相关系数均在该区间内则表示该变量自相关性问题不严重。由自相关图可以看出序列 0 阶自相关，因此 MA 的阶数 $q=0$。

当滞后期取 lag=50 时，由偏自相关图可以确定 $p=8,16,24$（见图 7.4）。

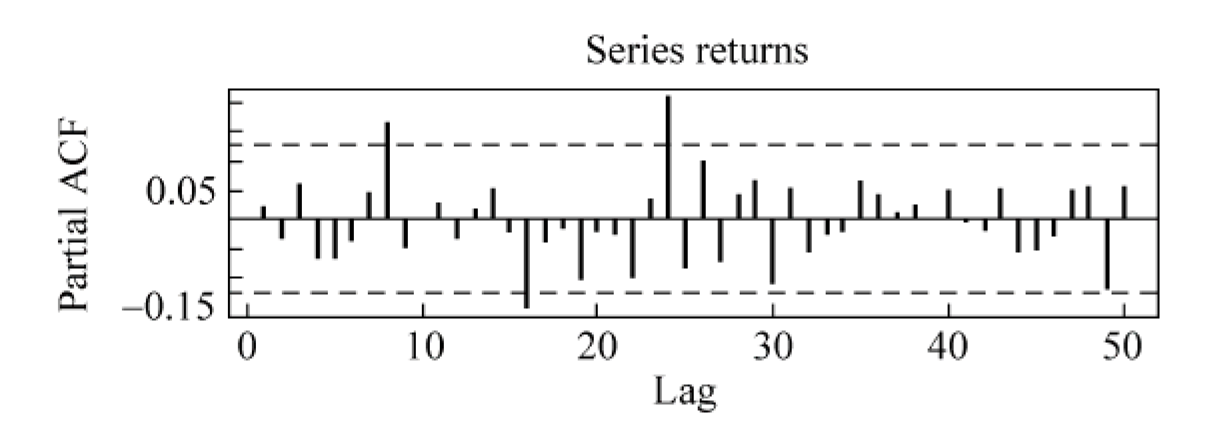


图 7.4　偏自相关函数图

7.2.3　ARMA 模型拟合

```
> ARMAfit <- arma(returns, lag = list(ar = c(8,16,24), ma = NULL), series = "returns")
> summary(ARMAfit)
Call:
arma(x = returns, lag = list(ar = c(8, 16, 24), ma = NULL), series = "returns")
Model:
ARMA(24,0)
Residuals:
       Min         1Q       Median        3Q        Max
 -0.0334173  -0.0084237  -0.0003561  0.0062042  0.0773497
Coefficient(s):
               Estimate      Std. Error     t value Pr(>|t|)
ar8            0.2059246     0.0601006       3.426 0.000612 ***
ar16          -0.2006215     0.0593793      -3.379 0.000728 ***
ar24           0.2291942     0.0592183       3.870 0.000109 ***
intercept     -0.0003592     0.0008163      -0.440 0.659935
 ---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Fit:
sigma^2 estimated as 0.0001608, Conditional Sum-of-Squares = 0.03, AIC = -1413.29
```

由上面拟合结果可以看出，ar8，ar16，ar24 阶系数均显著，因此可得 ARMA 模型表达式如下：

$$\hat{r}_1 = -0.00036 + 0.20592 r_{t-8} - 0.20062 r_{t-16} + 0.22919_{t-24}$$

带入相应滞后收益率即可实现未来收益率的预测。

7.3 GARCH 模型

GARCH(Generalized ARCH)即广义 ARCH 模型，是 Bollerslev 在 1986 年提出的。ARCH(Auto-Regressive Condition Heteroskedasticity)模型即自回归条件异方差模型，是 Engle 在 1982 年提出的，是一种典型的金融时间序列波动性分析模型。该模型具有计算时间序列的条件方差的特点，能够较准确地刻画金融时间序列的特征，包括波动性、收益率的不相关性等。但 ARCH 模型要求条件方差必须为正值，因而需要更具一般适用性的模型。GARCH 模型是根据过去方差及其预测值进行方差预测的，对波动性的分析及预测具有更好的效果，具有广泛的理论及实际应用价值。

7.3.1 GARCH 模型简介

一般的 GARCH 模型可以表示为

$$r_t = c_1 + \sum_{i=1}^{R} \phi_i r_{t-i} + \sum_{j=1}^{M} \phi_j \theta_{t-j} + \theta_t \tag{7-1}$$

$$\theta_t = u_t \sqrt{h_t} \tag{7-2}$$

$$h_t = k + \sum_{i=1}^{q} G_i h_{t-i} + \sum_{i=1}^{p} A_i \theta_{t-i}^2 \tag{7-3}$$

其中，h_t 为条件方差，u_t 为独立同分布的随机变量，h_t 与 u_t 互相独立，u_t 为标准正态分布。式(7-1)称为条件均值方程；式(7-3)称为条件方差方程，说明时间序列条件方差的变化特征。

7.3.2 GARCH 模型拟合

GARCH(1,1)模型是 GARCH 模型中简单且应用广泛的一个模型。运用 R 对 7.1.1 节中的收益率时间序列进行建模分析如下：

```
> library(fGarch)
> Garchfit <- garchFit(~ garch(1,1), data = returns)
> summary(Garchfit)
Title:
GARCH Modelling
Call:
garchFit(formula = ~garch(1, 1), data = returns)
Mean and Variance Equation:
data ~ garch(1, 1)
<environment: 0x054fe368>
[data = returns]
```

```
Conditional Distribution:
norm
Coefficient(s):
          mu        omega         alpha1       beta1
1.4177e-04  1.9869e-06  1.8400e-02  9.6909e-01
Std. Errors:
based on Hessian
Error Analysis:
          Estimate    Std. Error    t value Pr(>|t|)
mu        1.418e-04  1.000e-03    0.142   0.887
omega     1.987e-06  4.682e-06    0.424   0.671
alpha1    1.840e-02  2.035e-02    0.904   0.366
beta1     9.691e-01  2.049e-02    47.288   <2e-16 ***
Signif. codes: 0 '***'0.001 '**'0.01 '*'0.05 '.'0.1 ''1
Log Likelihood:
693.9436      normalized: 2.879434
Standardised Residuals Tests:
                                Statistic p-Value
Jarque-Bera Test    R     Chi^2     339.9752     0
Shapiro-Wilk Test   R     W         0.9405965    2.579529e-08
Ljung-Box Test      R     Q(10)     11.16604     0.3447291
Ljung-Box Test      R     Q(15)     13.49275     0.5642944
Ljung-Box Test      R     Q(20)     20.06933     0.4536006
Ljung-Box Test      R^2   Q(10)     4.317861     0.9318706
Ljung-Box Test      R^2   Q(15)     6.802526     0.9628995
Ljung-Box Test      R^2   Q(20)     7.938161     0.992269
LM Arch Test        R     TR^2      5.243007     0.9493658
Information Criterion Statistics:
      AIC        BIC          SIC        HQIC
-5.725673  -5.667834  -5.726212  -5.702371
```

由 GARCH 模型可以获得波动率序列。

```
>Valat<-volatility(Garchfit)
>head(Valat)      # 查看前 6 个波动率数据
[1] 0.01362099 0.01351810 0.01345310 0.01333020 0.01322306 0.01310385
```

在 GARCH 模型拟合时，可以通过增设参数 cond.dist = c("norm", "snorm", "ged", "sged", "std", "sstd", "snig", "QMLE")实现不同分布下的 GARCH 模型拟合，如正态分布、广义误差分布、偏 t 分布等，默认状态为正态分布拟合。由上面拟合结果可以得到波动率预测公式：

$$\sigma_t^2 = 0.000002 + 0.01814 r_t^2 + 0.9691 \sigma_{t-1}^2$$

GARCH 模型常被用作股市价格波动性分析，由计算的波动率可以进一步分析股市波动带来的风险影响，例如可以进行风险价值(VaR)的计算，正态分布下 k 期 VaR 值为：

$$\mathrm{VaR}_t = \sqrt{k}\sigma_t$$

将 GARCH 模型计算出的波动率代入上式，即可实现风险价值 VaR 的计算。

CHAPTER 8

第8章 动态面板数据专题

如果仅仅是分析横截面数据或者时间序列数据，不能全面地反映复杂的经济现象，而动态面板数据融合了截面数据和时间序列数据，使得数据信息得到充分利用，从而得到更为有效的参数估计。本章介绍了动态面板数据模型 GMM 估计方法的原理，并给出了一个动态面板数据实例。

8.1 GMM 估计

8.1.1 系统 GMM 估计

广义矩估计（Generalized Method of Moments，GMM）是基于模型实际参数满足一定矩条件而形成的一种参数估计方法，是矩估计方法的一般化。传统的计量经济学估计方法，例如普通最小二乘法、工具变量法和极大似然法等都存在自身的局限性。即其参数估计量必须在满足某些假设时，比如模型的随机误差项服从正态分布或某一已知分布时才是可靠的估计量。而 GMM 不需要知道随机误差项的准确分布信息，允许随机误差项存在异方差和序列相关，因而所得到的参数估计量比其他参数估计方法更有效。因此，GMM 方法在模型参数估计中得到广泛应用。

系统 GMM 估计方法可以控制模型中可能存在的内生性和异方差问题。该方法对估计模型进行一阶差分，将弱外生变量的滞后项作为相应变量的工具变量，从而获得一致有效的估计，可以避免严重的有限样本误差。而其他方法如混合最小二乘法、固定效应模型等由于只是简单地做了解释变量与误差项的协方差为零、不存在异方差等假设而难以达到较好的估计效果。而实际中，解释变量一般具有内生性，可能同时决定被解释变量；误差项具有序列相关性，并非独立同分布，故而会产生有偏的、不一致的估计结果，所得出的参数含义可能出现误导。

8.1.2 GMM 估计原理

在动态面板数据模型中，由于因变量滞后项作为解释变量，从而有可能导致解释变量与随机扰动项相关，且模型具有横截面相依性。因而，传统估计方法进行估计时必将产生参数估计的有偏性和非一致性，从而使根据参数而推断的经济学含义发生扭曲。针对以上情况，Arellano 和 Bond(1991)，Blundell 和 Bond(1998)提出 GMM 估计很好地解决了上述问题。以下列形式的动态面板数据模型为例简要说明 GMM 估计的基本原理。

(1) 建立动态面板数据模型。

$$Y_{it} = \alpha_0 Y_{it-1} + \sum \alpha_i X_{it} + \varepsilon_{it} \tag{8-1}$$

其中，Y_i 为被解释变量，X_i 为解释变量，α_0、α_i 为待估系数，ε_i 为随机误差项。

(2) GMM 估计的首要条件是运用工具变量产生相应的矩条件方程。为此，对式(8-1)进行一阶差分得到式(8-2)。

$$\Delta Y_{it} = \alpha_0 \Delta Y_{it-1} + \sum \alpha_i \Delta X_{it} + \Delta\varepsilon_{it} \tag{8-2}$$

可得残差表达式

$$\Delta\varepsilon_{it}(\alpha) = \Delta Y_{it} - \alpha_0 \Delta Y_{i(t-1)} - \sum \alpha_i \Delta X_{it} \tag{8-3}$$

(3) 对式(8-1)进行一阶差分是为达到时间平稳效应，主要目的在于选取合适的工具变量和产生相应的矩条件方程。通常将 Y_{it-2}，Y_{it-3} 作为工具变量。设 $f(\alpha)$ 为矩条件方程，有

$$f(\alpha) = \sum f_i(\alpha) = \sum \boldsymbol{Z}_i \Delta\varepsilon_i(\alpha) \tag{8-4}$$

其中，$\boldsymbol{Z}_i$ 为所选取的工具变量向量。

GMM 估计的基本思想是选择使样本矩之间的加权距离最小，即 GMM 的估计量是目标函数极小化时的参数估计量。

$$\min S(\alpha) = f'(\alpha) H f(\alpha) = \left[\sum \boldsymbol{Z}_i \Delta\varepsilon_i(\alpha)\right]' H \left[\sum \boldsymbol{Z}_i \Delta\varepsilon_i(\alpha)\right] \tag{8-5}$$

其中，H 为所选取的权重矩阵。

8.2 动态面板数据模型的系统 GMM 估计

1. 数据分析

(1) 数据统计描述。进行数据分析之前对数据的统计特征进行描述，结果如表 8.1 所示，运行代码如下：

```
 # 0. 初始化
> setwd('E:/R/RCode/SYS GMM')
> rm(list = ls())
 # 1. 读取原始数据
 ## (1) 取出数据中的 Firm 文本列
> Firm_LLEV <- read.table("clipboard",colClass = "character")
                                        # 打开 Excel 表复制文本列，运行该语句
> Firm_LEV <- read.table("clipboard",colClass = "character")
> save(Firm_LLEV, Firm_LEV, file = "Firm.RData")  #将文本列保存为 R 数据文件
> load("Firm.RData")                              # 以后使用可直接加载"文本列"
```

```
## (2) 读取 Excel 格式的数据
> library(RODBC)                        # 加载包
> CS_data <- odbcConnectExcel("CapStruct.xls")
> LLEV <- sqlFetch(CS_data, "LLEV")
> LEV <- sqlFetch(CS_data, "LEV")
> close(CS_data)
## (3) 整理读入的数据
> LLEV[,1] <- Firm_LLEV[-1,]            # 将 Firm 列的股票代码还原为文本格式
> LEV[,1] <- Firm_LEV[-1,]
> head(LLEV)                            # 查看数据的前 6 行
> head(LEV)
> LEV <- LEV[,-13:(-14)]                # 删除后面的空格两列
> save(LLEV, LEV, file="GMMData.RData") # 原始数据集已完全整理为 R 数据文件并保存
> load("GMMData.RData")
# 2. 数据的基本特征
## (1) summary 函数特征描述
> summary(LLEV)
> summary(LEV)
## (2) describe 函数特征描述
> library(Hmisc)
> describe(LLEV)   # 如果 unique<10,那么该变量是离散的; 如果 uinque>20,那么会输出 5 个
最低和最高的值
> describe(LEV)
> dim(LLEV)
> class(LLEV)
> dim(LEV)
> class(LEV)
> LLEV_FirmNum <- unique(LLEV$Firm)
> length(LLEV_FirmNum)
> LEV_FirmNum <- unique(LEV$Firm)
> length(LEV_FirmNum)
```

表 8.1 数据统计描述

统 计 描 述	最小值	均值	中位数	最大值
overall Leverage (LED)	0.0000	0.7532	0.5176	877.2559
Long-term leverage (LTLV)	0.0000	0.0709	0.0263	1.8917
ROA	−2146.161	2.159	0.029	23509.769
Growth opportunities	−90989.00	−4.87	0.90	10860.78
Asset tangibility	0.0000	0.4638	0.4608	0.9946
Size	10.84	21.37	21.25	30.10
Earnings Volatitity	0.0	891.3	86.7	1080602.0
Non-debt tax shields	−0.03366	0.05704	0.02224	163.76950
Independent directors	0.0000	0.3515	0.3333	0.7500
Dummy CEO 1	0.0000	0.1386	0.0000	1.0000
'Ownership concentration	0.82	38.41	36.17	98.86

(2) 缺失数据处理。在建立模型之前，对所选取的数据进行了缺失值处理。首先，识别缺失数据，检查缺失数据，探究缺失数据模式。图 8.1 以图形方式展示了不同变量存在的缺失数据。

```
## (3) 缺失值 NA 处理
> library(mice)
> LLEV_Na.Pattern <- md.pattern(LLEV[, -1:(-2)])      # 展示缺失值模式的表格
> LEV_Na.Pattern <- md.pattern(LEV[, -1:(-2)])
> library(VIM)
> aggr(LLEV[, -1:(-2)],prop = FALSE,numbers = TRUE)     # 图形探究缺失数据
> aggr(LEV[, -1:(-2)], prop = FALSE,numbers = TRUE)
> LLEV_Na.ind <- which(is.na(LLEV), arr.ind = TRUE)     # 缺失值下标矩阵
> LEV_Na.ind <- which(is.na(LEV), arr.ind = TRUE)
> LLEV_Varind <- as.data.frame(which(is.na(LLEV[, -3]), arr.ind = TRUE))
# 解释变量缺失值下标矩阵
> LEV_Varind <- as.data.frame(which(is.na(LEV[, -3]), arr.ind = TRUE))
> LLEV_Nna <- LLEV[ - LLEV_Varind$row, ]
> LEV_Nna <- LEV[ - LEV_Varind$row, ]
> dim(LLEV_Nna)
> class(LLEV_Nna)
> dim(LEV_Nna)
> class(LEV_Nna)
> LLEV_Nna_FirmNum <- unique(LLEV_Nna$Firm) # 统计剔除 NA 值后的 Firm 个数
> dim(LLEV_Nna_FirmNum)
> LEV_Nna_FirmNum <- unique(LEV_Nna$Firm)
> length(LEV_Nna_FirmNum)
```

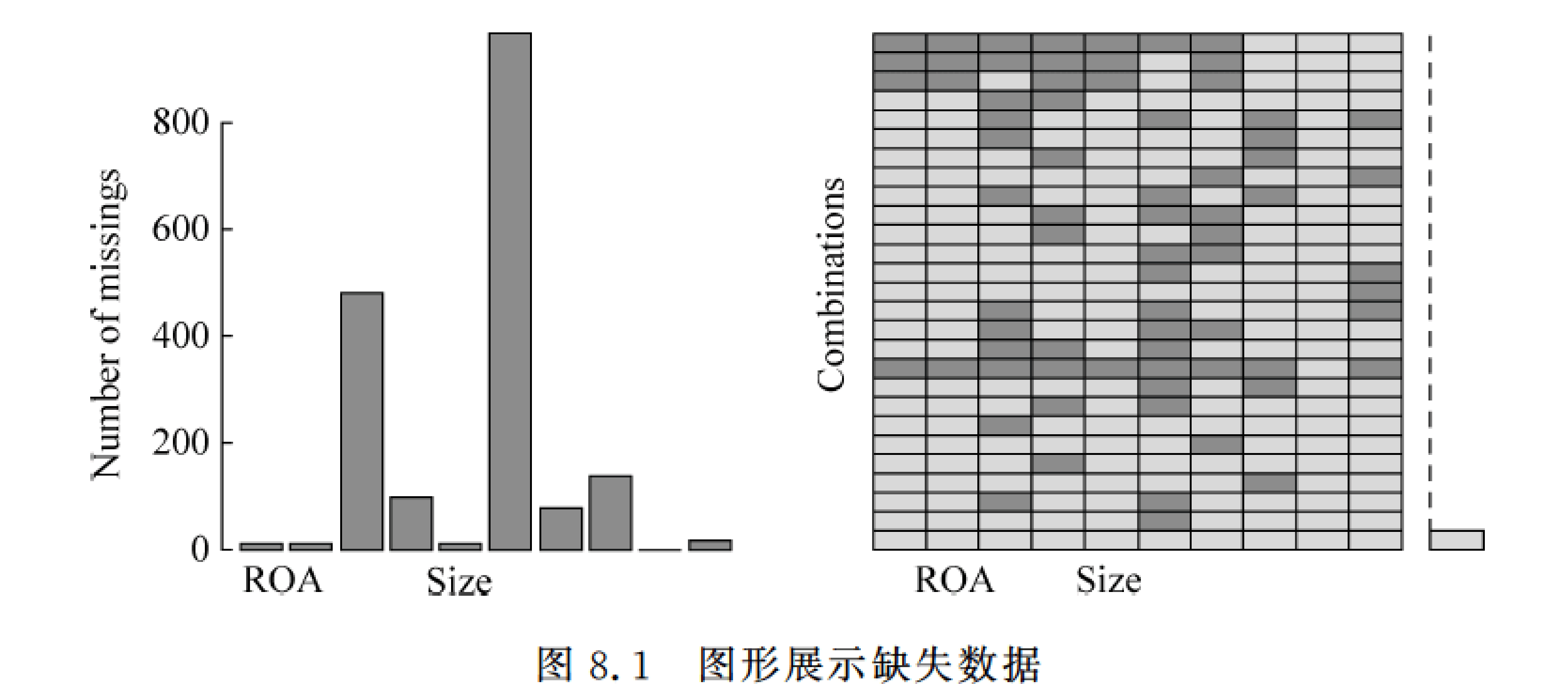


图 8.1　图形展示缺失数据

一般地，被解释变量是根据解释变量去预测，所以当解释变量存在缺失时无法进行被解释变量的预测。因此，此处缺失数据的处理标准为如果解释变量存在某个缺失值，那么将相应行数据全部进行剔除。

2. 模型选择

经济理论告诉我们，杠杆效应是一个连续动态的过程，上期的杠杆效应对当期杠杆效应产生某种影响，因而引入滞后因变量更符合理论与现实。然而一旦将滞后因变量引入方程，原本的静态模型将会转变为动态模型，现有的一般估计方法将会失效，结论的准确性也将无

从保证，只有采用动态面板数据模型(DPD)才能进行较为有效的估计。为此，本文选用行业面板数据，在研究方法上采用了基于广义矩估计的动态面板数据模型。

为研究总杠杆效应(Overall Leverage，LED)、长期杠杆效应(Long-Term Leverage，LTLV)分别受哪些变量的影响，可分别将LED、LTLV设定为被解释变量，并分别选取9种变量作为解释变量：ROA、Growth opportunities、Asset tangibility、SIZE、Earnings Volatility、Non-debt tax shield、Independent director、Dummy CEO 1、Ownership concentration，根据动态面板数据模型一般表达式分别建立两种动态面板数据模型。

模型一：设LED为被解释变量，解释变量除选取上述9种变量之外，另选取LED的滞后1阶，ROA、Growth Opportunities、SIZE、Non-Debt Tax Shield 的滞后1阶、2阶作为解释变量，被解释变量LED的滞后3阶作为工具变量。

$$Y_{it}^{LED} = \alpha_0 Y_{i(t-1)}^{LED} + \sum_{i=1}^{17} \alpha_i X_{it} + \varepsilon_{it} \tag{8-6}$$

模型二：设LTLV为被解释变量，解释变量除选取上述9种变量之外，另选取LTLV的滞后1阶、2阶，Growth Opportunities、SIZE、Earnings Volatility、Ownership concentration 的滞后1阶作为解释变量，被解释变量LTLV的滞后3～8阶作为工具变量。

$$Y_{it}^{LTLV} = \beta_0 Y_{i(t-1)}^{LTLV} + \sum_{i=1}^{15} \beta_i X_{it} + \varepsilon_{it} \tag{8-7}$$

在系统GMM估计中存在动态模型设定是否适当和工具变量选择是否有效的问题。判断的技术标准有两个。

(1) Sargan检验，也称为J检验。Sargan检验用于检验是否存在过度识别。在原假设成立的条件下(无过度识别)，渐近服从卡方分布，自由度为工具个数与参数个数之差。

(2) 模型差分的残差是否序列相关。运用模型1阶差分的残差 n 阶序列相关的统计量 $m(n)$ 来判断工具变量的有效性，$m(n)$ 的原假设为无序列相关，渐近服从正态分布。基于此策略选择工具变量，如果差分残差存在1阶序列相关，工具变量必须取滞后2阶或更高阶才有效；依此类推，如果2阶序列相关，工具变量须为滞后3阶或更高阶(Brown 和 Petesen，2009)。

由动态系统GMM估计步骤分别得到两模型的参数估计结果如表8.2所示。代码如下：

```
# 3. 系统 GMM 模型
## (1)加载面板数据模型包
> library(plm)
## (2)将原始数据转化为面板模型数据
> LLEV_data <- pdata.frame(LLEV_Nna, c("Firm ","Year"), drop = TRUE)
> LEV_data <- pdata.frame(LEV_Nna, c("Firm ","Year"), drop = TRUE)
## (3)建立模型
> colnames(LLEV_data) <- c("Yllev","Roa","Gopp","Asst","Size","Evol","Ndt"," Indd","
Dceo","Ocon") #给变量重新命名
> LLEV_SGMM <- pgmm(Yllev ~ lag(Yllev,1:2) + Roa + Gopp + lag(Gopp,1) + Asst + Size + lag
(Size,1) + Evol
                  + lag(Evol,1) + Ndt + Indd + Dceo + Ocon + lag(Ocon,1) | lag(Yllev, 3:8),
```

```
                 data = LLEV_data, effect = "twoways", model = "twosteps", transformation = "
ld")
> summary(LLEV_SGMM, robust = TRUE)
> colnames(LEV_data) <- c("Ylev","Roa","Gopp","Asst","Size","Evol","Ndt","Indd","Dceo","
Ocon") #给变量重新命名
> LEV_SGMM <- pgmm(Ylev ~ lag(Ylev,1) + Roa + lag(Roa,1:2) + Gopp + lag(Gopp,1:2) + Asst +
Size + lag(Size,1:2)
                    + Evol + Ndt + lag(Ndt,1:2) + Indd + Dceo + Ocon| lag(Ylev,3),
                 data = LEV_data, effect = "twoways", model = "twosteps", transformation = "
ld")
> summary(LEV_SGMM, robust = TRUE)
```

表 8.2　系统 GMM 模型参数估计结果

系统 GMM 估计参数	模型一 LED	模型二 LTLV
Lag(LED,1)	1.9895e-01 (4.5534e-02) ***	____
lag(LTLV, 1)	____	6.2063e-01 (3.0276e-02) ***
lag(LTLV, 2)	____	−4.9580e-02 (4.0067e-02)
ROA	−3.1174e-02 (1.4411e-04) ***	7.2820e-07 (5.7126e-07)
lag(ROA,1)	7.9660e-03 (1.4184e-03) ***	____
lag(ROA,2)	−6.1856e-01 (8.9142e-01)	____
Growth opportunities	−5.9986e-06 (1.8450e-05)	−2.3664e-06 (2.0525e-06)
lag(Growth opportunities,1)	6.1204e-06 (1.0918e-06) ***	3.7564e-07 (1.1031e-07) ***
lag(Growth opportunities,2)	4.3835e-06 (1.6768e-06) **	____
Asset tangibility	−8.2201e-02 (8.3999e-02)	3.1781e-02 (8.7562e-03) ***
Size	−2.4352e-01 (1.6859e-01)	3.3865e-02 (3.8240e-03) ***
lag(Size,1)	1.6429e-01 (1.5382e-01)	−2.2694e-02 (3.6209e-03) ***
lag(Size,2)	9.9201e-02 (7.4498e-02)	____
Earnings Volatitity	−4.9290e-08 (2.0200e-07)	8.1813e-09 (1.1842e-08)
lag(Earnings Volatitity,1)	____	−2.7945e-08 (1.2320e-08) *
Non-debt tax shields	5.3665e+00 (1.7496e-02) ***	1.7831e-03 (2.7154e-04) ***
lag(Non-debt tax shields,1)	−4.9273e-01 (2.4635e-01) *	____
lag(Non-debt tax shields,2)	−4.9635e+00 (1.4704e+00) ***	____
Independent directors	−3.2618e-01(2.7035e-01)	7.6915e-03 (2.5750e-02)
Dummy CEO 1	4.0240e-02(4.0133e-02)	−1.2759e-03 (3.1224e-03)
'Ownership concentration	−1.4356e-03(1.4422e-03)	2.4383e-04 (1.9514e-04)
lag('Ownership concentration,1)	____	−3.5256e-04(1.8561e-04)
J (p-value)	36.4244(0.0654)	36.2445(0.0873)
m1 (p-value)	−0.3530 (0.7241)	−4.9510 (0.0000)
m2 (p-value)	−0.7253 (0.4683)	0.3577(0.7205)
Firms/observations	1584/9061	1584/9061

注：模型参数估计值后“()”中的数值是相应估计量的标准差；*、** 和 *** 分别表示在 10%、5%和 1%的显著水平下模型系数是否显著；“____”表示相应模型中没有采用该参数变量。

由表 8.2 估计结果，可以有如下结论：

(1) J 统计量及 P 值显示，模型一、模型二接受原假设，均不存在过度识别问题，因而工

具变量选择适当。

(2) 由 m(2)统计量及 P 值可知，模型一、模型二一阶差分残差的二阶序列自相关检验均接受原假设，即不存在序列相关性，渐近服从正态分布。

由以上分析可知，动态模型一、模型二的系统 GMM 估计满足其两个判断技术标准，因此模型的工具变量选择有效，模型设定合理。

3. 结果分析

OLS 没有考虑误差项构成，只是简单地假设解释变量与误差项不相关。而实际上解释变量可能被同时决定，即具有内生性。固定效应(FE)虽然可以控制未观察到的特定企业异质性，但同样不能控制内生性问题。而且 OLS、FE 还假设不存在异方差。由于这些假设有违现实，因此其估计结果是有偏差的，而 GMM 估计正好能够解决这些现实问题。

CHAPTER 9

第9章

数据挖掘专题

数据挖掘(Data Mining,DM)指的是从数据中挖掘出有用信息的过程。数据挖掘是一个非常热门的专题,本章介绍几种典型的数据挖掘方法,包括关联规则、贝叶斯分类、决策树、人工神经网络、支持向量机等算法,并给出在R环境下实现相应算法的应用实例。

9.1 关联规则

关联规则是数据挖掘中一个重要的算法,它能挖掘得到变量之间的依赖关系,给决策提供一定的依据,在实际中有着较为广泛的应用。例如,在商品交易中,关联规则是发现不同商品之间的联系,得到顾客购买行为模式,比如顾客会一起购买的商品有哪些,根据这些模式可以进行商品的摆放设计及顾客的分类等。一个典型的例子就是尿布与啤酒的购买模式,即和尿布一起购买最多的商品是啤酒,根据这一关联规则可以将尿布和啤酒摆放在同一货架上。规则中的项(Item)指的是表中非主键及外键属性的取值,每一个取值为一个项。项集(Itemset),即项组成的集合,记为 $I=\{i_1,i_2,\cdots,i_m\}$。关联规则指的是类似于 $A\rightarrow B$ 的蕴涵式,其中 $A\subset I$,$B\subset I$ 且 A 和 B 不相交。支持度和可信度是关联规则中重要的两个概念。规则的支持度 $S(A\rightarrow B)=P(AB)=\frac{|AB|}{|D|}$,即数据库 D 中事务同时包含 AB 的概率。规则的可信度 $C(A\rightarrow B)=P(B|A)=\frac{|AB|}{|A|}$,即包含项集 A 的同时也包含 B 的条件概率。要挖掘出有用的关联规则需要设定阈值,即最小支持度和最小可信度。频繁项集指的是满足最小支持度的项集。若规则同时满足最小支持度和最小可信度,则为关联规则或强关联规则。兴趣度 $I(A\rightarrow B)=\frac{P(AB)}{P(A)P(B)}$,刻画了 A 与 B 的相关程度,若兴趣度等于1,则 A 和 B 的出现是相互独立的;若兴趣度大于1,则 A 和 B 是正相关的,这种情况下兴趣度越大,关联规则越具有实际应用价值;反之,若兴趣度小于1,则 A 和 B 是负相关的,此时兴趣度越小,规则的反面规则越具有实际应用意义。

Apriori 算法是一种经典的关联规则算法，该算法是按照一定规则生成候选频繁集进而找到频繁模式。该算法将关联规则的挖掘分为以下两个阶段进行，第一阶段：找出所有的频繁项集，即支持度大于最小支持度的项集；第二阶段：运用第一阶段找出的频繁项集得到关联规则。

在 R 环境中实现 Apriori 算法如下：

1. 加载包

```
> library(arules)
```

2. 载入 transaction 数据对象 Adult

```
> data(Adult)
```

3. 输出关联规则对象 rules

apriori 函数接受一个 transaction 对象的输入，输出关联规则对象 rules。为方便起见，这里用于计算的 transaction 对象 Adult 是通过第 5 行从 arules 包中现成载入进来的。

```
> rules <- apriori(Adult, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))
parameter specification:
confidence minval smax arem aval originalSupport support minlen maxlen
        0.9    0.1    1 none FALSE      TRUE    0.5    1    10
  target ext
  rules FALSE
algorithmic control:
  filter tree heap memopt load sort verbose
    0.1 TRUE TRUE FALSE TRUE    2    TRUE
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09)    (c) 1996 - 2004 Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[115 item(s), 48842 transaction(s)] done [0.06s].
sorting and recoding items ... [9 item(s)] done [0.01s].
creating transaction tree ... done [0.07s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [52 rule(s)] done [0.00s].
creating S4 object ... done [0.01s].
```

9.2 降维分析

多维标度分析（MDS）是一种将多维空间的研究对象简化到低维空间进行定位、分析和归类，同时又保留对象间原始关系的数据分析方法。

设想一下，如果在欧氏空间中已知一些点的坐标，由此可以求出欧氏距离。那么反过

来，已知距离应该也能得到这些点之间的关系。这种距离可以是古典的欧氏距离，也可以是广义上的“距离”。MDS就是在尽量保持这种高维度“距离”的同时，将数据在低维度上展现出来。

在经典MDS中，距离是数值数据表示，将其看作是欧氏距离。在R中stats包的cmdscale函数实现了经典MDS。它是根据各点的欧氏距离，在低维空间中寻找各点坐标而尽量保持距离不变。

(1) 下载数据。从 http://rosetta.reltech.org/TC/v15/Mapping/data/dist－Aus.csv 上下载 Australia 的 8 个城市间的距离数据。

```
> url <- "http://rosetta.reltech.org/TC/v15/Mapping/data/dist-Aus.csv"
> dist.au <- read.csv(url)
> dist.au
   X     A    AS     B     D     H     M     P     S
1  A     0  1328  1600  2616  1161   653  2130  1161
2  AS 1328     0  1962  1289  2463  1889  1991  2026
3  B  1600  1962     0  2846  1788  1374  3604   732
4  D  2616  1289  2846     0  3734  3146  2652  3146
5  H  1161  2463  1788  3734     0   598  3008  1057
6  M   653  1889  1374  3146   598     0  2720   713
7  P  2130  1991  3604  2652  3008  2720     0  3288
8  S  1161  2026   732  3146  1057   713  3288     0
```

(2) 移除第1列，将城市名称首写字母设置为行名称。

```
> row.names(dist.au) <- dist.au[, 1]
> dist.au <- dist.au[, -1]
> dist.au
      A    AS     B     D     H     M     P     S
A     0  1328  1600  2616  1161   653  2130  1161
AS 1328     0  1962  1289  2463  1889  1991  2026
B  1600  1962     0  2846  1788  1374  3604   732
D  2616  1289  2846     0  3734  3146  2652  3146
H  1161  2463  1788  3734     0   598  3008  1057
M   653  1889  1374  3146   598     0  2720   713
P  2130  1991  3604  2652  3008  2720     0  3288
S  1161  2026   732  3146  1057   713  3288     0
```

(3) cmdscale()降维。

```
> fit <- cmdscale(dist.au, eig = TRUE, k = 2)
> x <- fit$points[, 1]
> y <- fit$points[, 2]
```

k 表示数据的最大空间维度，eig 表示是否返回特征值，x、y 表示横轴、纵轴。

（4）可视化结果：在一张图上显示各城市的位置（图 9.1）。

```
> plot(x, y, pch = 19, xlim = range(x) + c(0, 600))
city.names <- c("Adelaide", "Alice Springs", "Brisbane", "Darwin", "Hobart",
    "Melbourne", "Perth", "Sydney")
> text(x, y, pos = 4, labels = city.names)
```

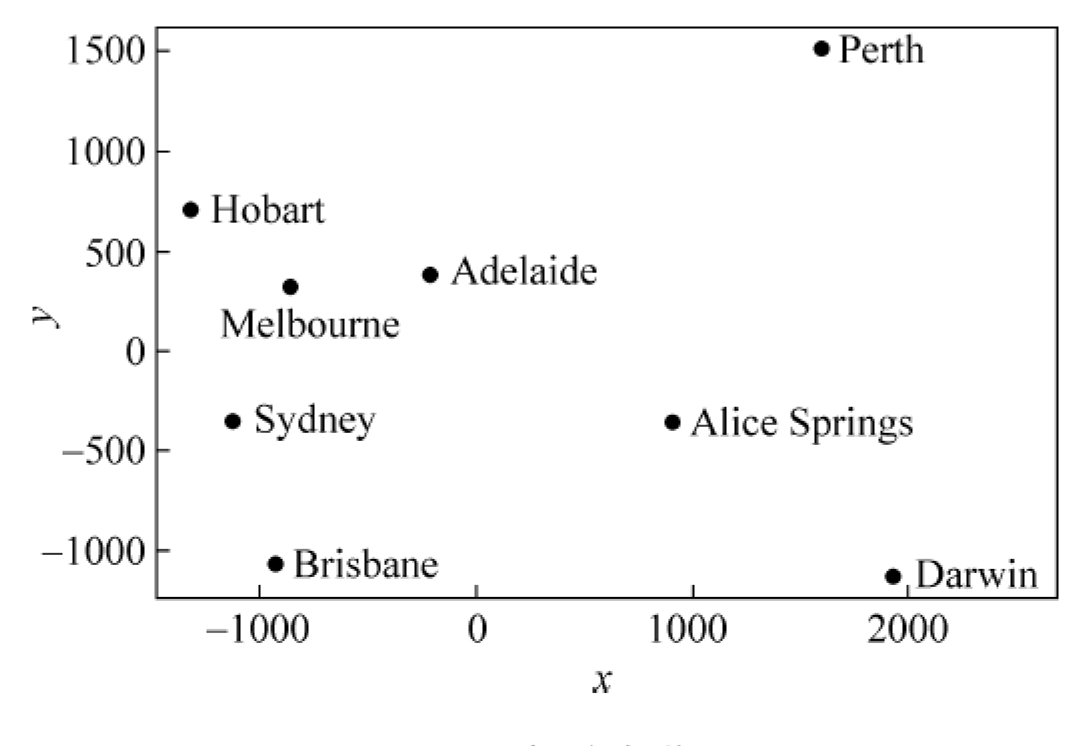


图 9.1　各城市位置

（5）翻转 x、y 轴。将 x、y 轴进行翻转，如图 9.2 所示，Darwin 和 Brisbane 则会移动到顶部（北部），这样方便在一张地图上对其进行比较。

```
> x <- 0 - x
> y <- 0 - y
> plot(x, y, pch = 19, xlim = range(x) + c(0, 600))
> text(x, y, pos = 4, labels = city.names)
```

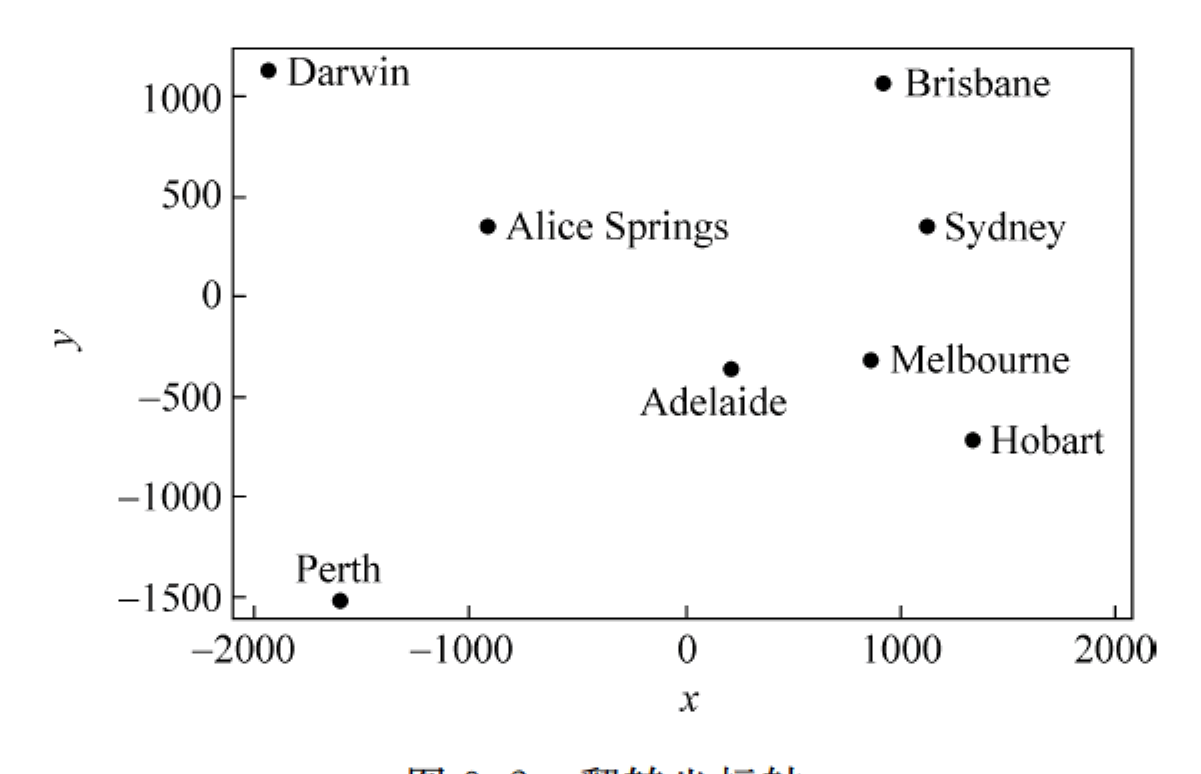


图 9.2　翻转坐标轴

（6）Igraph 包中 layout.mds()也可以实现降维。创建一个 8 个节点的图形，并设置布局为上述 8 个城市的距离矩阵，同样可以得到与图 9.2 类似的地图布局。

```
> library(igraph)
> g <- graph.full(nrow(dist.au))
> V(g) $ label <- city.names
> layout <- layout.mds(g, dist = as.matrix(dist.au))
> plot(g, layout = layout, vertex.size = 3)
```



图 9.3　可视化结果

9.3　社交网络分析

社交网络(Social Network),也称为社会网络,用来分析一组社会行动者之间关系结构特征的方法。社交网络正朝着节约社交时间和物质成本,获取高速、有效信息这一方向发展。社交网络通过网络这一平台把不同的人联系起来形成具有某一特点的团体。

社交网络理论认为社会是一个网络图,这个网络图由节点(Nodes)和节点与节点之间的连边组成。对于节点可以认为是各个社会行动者,在社交网络中任何一个社会单位或者社会实体可以看作是“点”,而社会连带则是各个行动者之间的社会关系,既可指朋友关系、同事关系等,也可指因分享经验、资源或信息而产生的互动关系。

(1) 数据下载。

```
> # load termDocMatrix
> load("data/termDocMatrix.rdata")
> # inspect part of the matrix
> termDocMatrix[5:10,1:20]
               Docs
Terms           1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
  data          1  1  0  0  2  0  0  0  0   0   1   2   1   1   1   0   1   0   0   0
  examples      0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0
  introduction 0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   1
  mining        0  0  0  0  0  0  0  0  0   0   0   1   1   0   1   0   0   0   0   0
  network       0  0  0  0  0  0  0  0  0   0   0   0   0   0   0   1   0   1   1   1
  package       0  0  0  1  1  0  0  0  0   0   0   1   0   0   0   0   0   0   0   0
```

(2) 注意到上述矩阵是一个标准矩阵，而不是文本挖掘框架下的 term-document 矩阵。在包 tm 里运行 term-document 矩阵代码，需要先进行下述转化：

```
> termDocMatrix <- as.matrix(termDocMatrix)
```

(3) 将数据转化为邻接矩阵。

```
> # change it to a Boolean matrix
> termDocMatrix[termDocMatrix >= 1] <- 1
> # transform into a term - term adjacency matrix
> termMatrix <- termDocMatrix % * % t(termDocMatrix)
> # inspect terms numbered 5 to 10
> termMatrix[5:10,5:10]
              Terms
Terms         data examples introduction mining network package
  data        53   5        2            34     0       7
  examples    5    17       2            5      2       2
  introduction 2   2        10           2      2       0
  mining      34   5        2            47     1       5
  network     0    2        2            1      17      1
  package     7    2        0            5      1       21
```

(4) 建立图形。建立一个 term-term 邻接矩阵，各行各列代表一个 term。此处使用 igraph 包中的 graph.adjacency()建立图形。

```
> library(igraph)
> # build a graph from the above matrix
> g <- graph.adjacency(termMatrix, weighted = T, mode = "undirected")
> # remove loops
> g <- simplify(g)
> # set labels and degrees of vertices
> V(g) $ label <- V(g) $ name
> V(g) $ degree <- degree(g)
```

(5) 绘制图形(图 9.4)。

```
> # set seed to make the layout reproducible
> set.seed(3952)
> layout1 <- layout.fruchterman.reingold(g)
> plot(g, layout = layout1)
```

(6) 优化图形输出效果。进一步设置顶点标签的尺寸，对重要的 term 加以突出，基于权重设置连接线的宽度和透明度。这在顶点和连接线众多时应用优势突出。在下面的代码中，顶点和连接线分别从 V()和 E()获得。函数 rgb(red, green, blue, alpha)定义颜色，alpha 定义透明度。同样可以绘制得到与上述图形布局类似的效果图(图 9.5)。

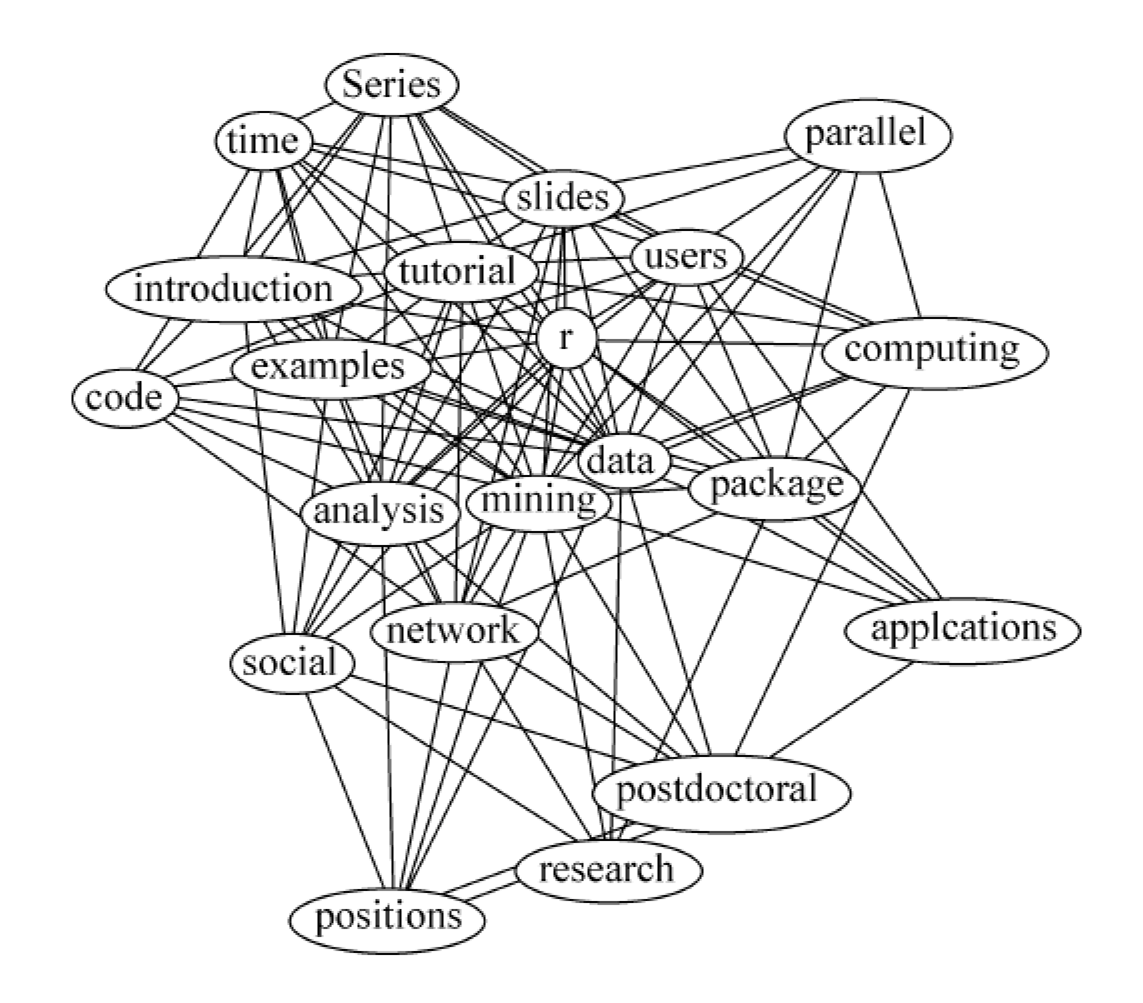


图 9.4　结果输出

```
> V(g) $ label.cex <- 2.2 * V(g) $ degree / max(V(g) $ degree) + .2
> V(g) $ label.color <- rgb(0, 0, .2, .8)
> V(g) $ frame.color <- NA
> egam <- (log(E(g) $ weight) + .4) / max(log(E(g) $ weight) + .4)
> E(g) $ color <- rgb(.5, .5, 0, egam)
> E(g) $ width <- egam
> # plot the graph in layout1
> plot(g, layout = layout1)
```

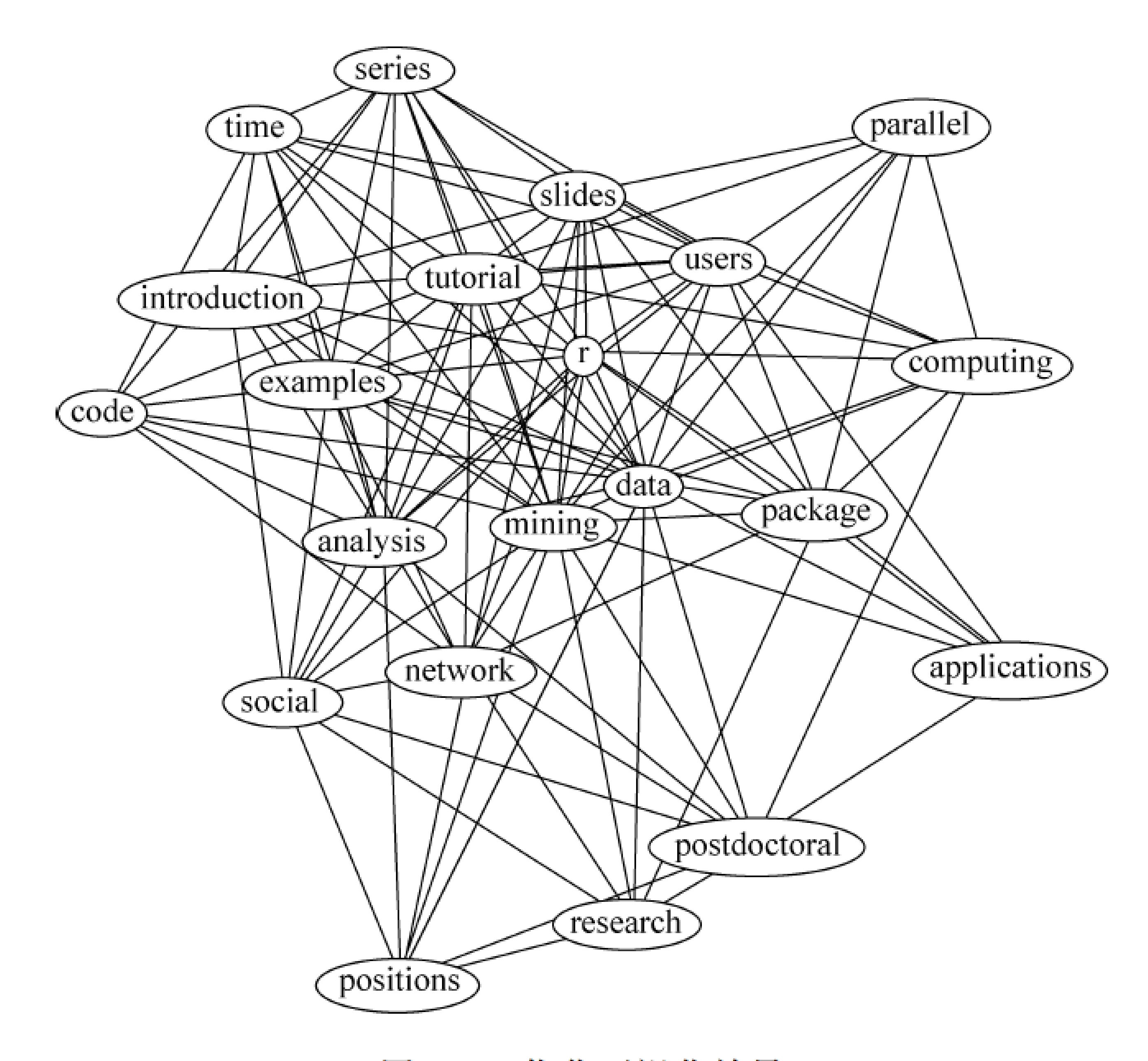


图 9.5　优化可视化效果

9.4 贝叶斯分类法

9.4.1 贝叶斯定理

贝叶斯定理涉及条件概率 $P(A|B)$ 指的是在事件 B 发生的条件下，事件 A 发生的概率，$P(A|B)=\frac{P(AB)}{P(B)}$。在实际问题中，人们常常能够观察得到 $P(A|B)$，但更想知道的是 $P(B|A)$，贝叶斯给出了一种根据前者计算后者的方法，即贝叶斯定理：$P(B|A)=\frac{P(A|B)P(B)}{P(A)}$。

贝叶斯定理通常应用于数据分类，其基本思想为对未知分类的数据记录，通过计算在该记录出现的条件下各个类别出现的概率，其中概率最大的类别即为此数据记录的类别。贝叶斯分类的步骤如下[①]。

(1) 假设 x 为未知分类的数据记录，记为 $x=\{a_1,a_2,\cdots,a_m\}$，其中 $a_1,a_2,\cdots,a_m$ 表示记录的特征属性。

(2) 假设 C 为类别的集合，记为 $C=\{y_1,y_2,\cdots,y_n\}$。

(3) 计算条件概率 $P(y_1|x),P(y_2|x),\cdots,P(y_n|x)$。

(4) 比较概率大小，确定类型：若有 $P(y_k|x)=\max\{P(y_1|x),P(y_2|x),\cdots,P(y_n|x)\}$，则待分类的数据记录 $x\in y_k$。

9.4.2 贝叶斯分类实例

下面在 R 环境下利用贝叶斯分类包 e1071 对鸢尾花(iris)数据集进行贝叶斯训练，并预测分类[②]。

(1) 安装并加载 e1071 包。

```
> install.packages("e1071")      # 安装包
> library(e1071)                 # 加载包
```

(2) 加载鸢尾花数据集。

包含 150 种鸢尾花的信息，其中每 50 种取自三个鸢尾花种之一(setosa、versicolour 及 virginica)。

```
> data(iris)
> iris
     Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1        5.1         3.5         1.4         0.2        etosa
2        4.9         3.0         1.4         0.2        setosa
3        4.7         3.2         1.3         0.2        setosa
```

① http://www.cnblogs.com/phoenixzq/p/3539619.html.
② http://my.oschina.net/letiantian/blog/324269? p=1.

```
4        4.6      3.1      1.5      0.2      setosa
5        5.0      3.6      1.4      0.2      setosa
6        5.4      3.9      1.7      0.4      setosa
7        4.6      3.4      1.4      0.3      setosa
8        5.0      3.4      1.5      0.2      setosa
9        4.4      2.9      1.4      0.2      setosa
10       4.9      3.1      1.5      0.1      setosa
```

以上显示的是 iris 数据集中前 10 条记录。该数据集中每个花的特征有以下 5 种属性描述：萼片长度(Sepal. Length)、萼片宽度(Sepal. Width)、花瓣长度(Petal. Length)、花瓣宽度(Petal. Width)、类别(Species)。

(3) 用鸢尾花数据集进行贝叶斯训练。

```
> classifier <- naiveBayes(iris[,1:4],iris[,5])
> classifier
Naive Bayes Classifier for Discrete Predictors
Call:
naiveBayes.default(x = iris[, 1:4], y = iris[, 5])
```

类别的先验概率：

```
A - priori probabilities:
iris[, 5]
    setosa versicolor virginica
0.3333333  0.3333333  0.3333333
```

特征 Sepal. Length 的条件概率(此处假设概率密度符合高斯分布)：

```
Conditional probabilities:
              Sepal.    Length
iris[, 5]     [,1]      [,2]
  setosa      5.006     0.3524897
  versicolor  5.936     0.5161711
  virginica   6.588     0.6358796
```

以上结果具体的解释为：对于特征 Sepal. Length，其中属于 setosa 类的概率符合 mean 为 5.006、标准差为 0.3524897 的高斯分布；属于 versicolor 类的概率符合 mean 为 5.936、标准差为 0.5161711 的高斯分布；属于 virginica 类的概率符合 mean 为 6.588、标准差为 0.6358796 的高斯分布。以下对于特征 Sepal. Width、Petal. Length 及 Petal. Width 的结果解释是一样的。

```
            Sepal.Width
iris[, 5]   [,1]      [,2]
  setosa    3.428     0.3790644
  versicolor 2.770    0.3137983
```

```
  virginica     2.974       0.3224966

             Petal.Length
iris[, 5]      [,1]        [,2]
  setosa        1.462       0.1736640
  versicolor    4.260       0.4699110
  virginica     5.552       0.5518947

             Petal.Width
iris[, 5]      [,1]        [,2]
  setosa        0.246       0.1053856
  versicolor    1.326       0.1977527
  virginica     2.026       0.2746501
```

（4）对鸢尾花数据集中第一条记录进行类别的预测。

```
> predict(classifier,iris[1, -5])
[1] setosa
Levels: setosa versicolor virginica
```

（5）该贝叶斯分类的效果。

```
> table(predict(classifier,iris[, -5]),iris[,5],dnn = list('predicted','actual'))
            actual
predicted    setosa versicolor virginica
  setosa        50         0         0
  versicolor    0         47        3
  virginica     0         3         47
```

由以上结果可以看出，分类该贝叶斯分类的分类预测效果很好。

（6）构造一条新记录并进行分类预测。

```
> new_data = data.frame(Sepal.Length = 7,Sepal.Width = 3,Petal.Length = 6,Petal.Width = 2)
> predict(classifier,new_data)
[1] virginica
Levels: setosa versicolor virginica
```

由此完成了运用 R 进行贝叶斯分类。

9.5 决策树

9.5.1 决策树原理

决策树方法起源于概念学习系统(Concept Learning System，CLS)，然后发展了 ID3 方法并达到高峰，最后又演化为能处理连续属性的 C4.5。有名的决策树方法还有 CART 和

Assistant。在机器学习中，决策树是一个预测模型，它代表的是对象属性与对象值之间的一种映射关系。

决策树提供了一种展示类似在什么条件下会得到什么值这类规则的方法。比如，在投保申请中要对投保风险的大小做出判断。决策树的基本组成部分为决策节点、分支和叶子。

决策树中最上面的节点称为根节点，是整个决策树的开始。决策树的每个节点子节点的个数与决策树所用的算法有关。如 CART 算法得到的决策树每个节点有两个分支，这种树称为二叉树。允许节点含有多于两个子节点的树称为多叉树。

每个分支要么是一个新的决策节点，要么是树的结尾，称为叶子。在沿着决策树从上到下遍历的过程中，在每个节点都会遇到一个问题，对每个节点上问题的不同回答导致不同的分支，最后会到达一个叶子节点。这个过程就是利用决策树进行分类的过程，利用几个变量(每个变量对应一个问题)来判断所属的类别(最后每个叶子会对应一个类别)。

建立决策树的过程，即树的生长过程是不断地把数据进行切分的过程，每次切分对应一个问题，也对应着一个节点。对每个切分都要求分成的组之间的“差异”最大。各种决策树算法之间的主要区别就是对这个“差异”衡量方式的区别。

9.5.2　决策树分类实例

下面在 R 环境下利用神经网络包 RSNNS 对鸢尾花(iris)数据集进行分类，并画出决策树。运行该例子前需先安装 RSNNS 包。

(1) 加载程序包，查看数据的结构。

```
> library("party")
> data(iris)
```

(2) 使用 ctree()创建决策树。

ctree()的第一个参数是方程设置，定义因变量和一系列自变量。

```
> iris_ctree <- ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width,
data = iris)
```

(3) 输出树。

```
> print(iris_ctree)
         Conditional inference tree with 4 terminal nodes
Response: Species
Inputs: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations: 150
1) Petal.Length <= 1.9; criterion = 1, statistic = 140.264
  2) * weights = 50
1) Petal.Length > 1.9
  3) Petal.Width <= 1.7; criterion = 1, statistic = 67.894
    4) Petal.Length <= 4.8; criterion = 0.999, statistic = 13.865
      5) * weights = 46
```

```
      4) Petal.Length > 4.8
        6) * weights = 8
    3) Petal.Width > 1.7
      7) * weights = 46
```

（4）绘制树（图 9.6）。

```
> plot(iris_ctree)
```

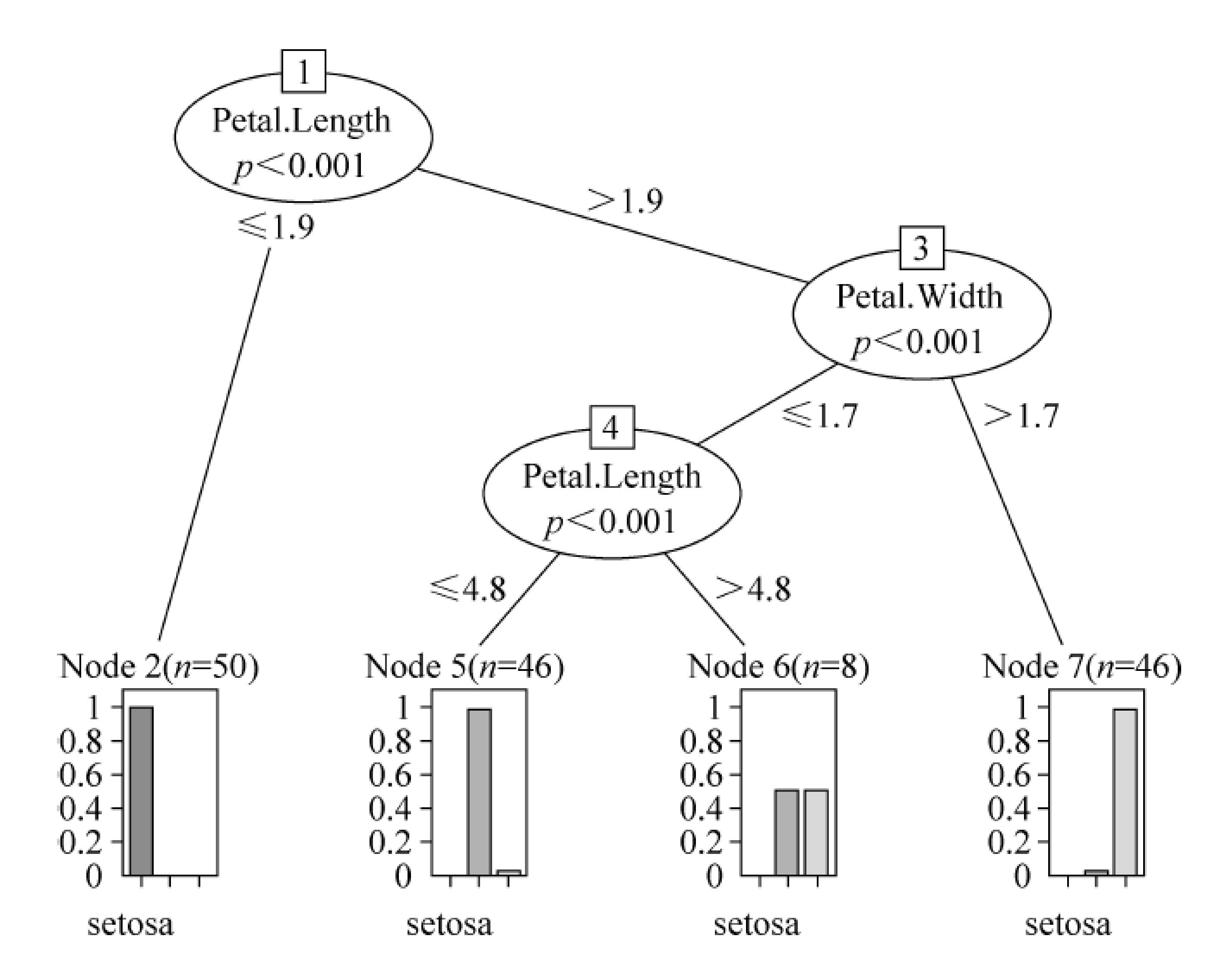


图 9.6　决策树

（5）简化树（图 9.7）。

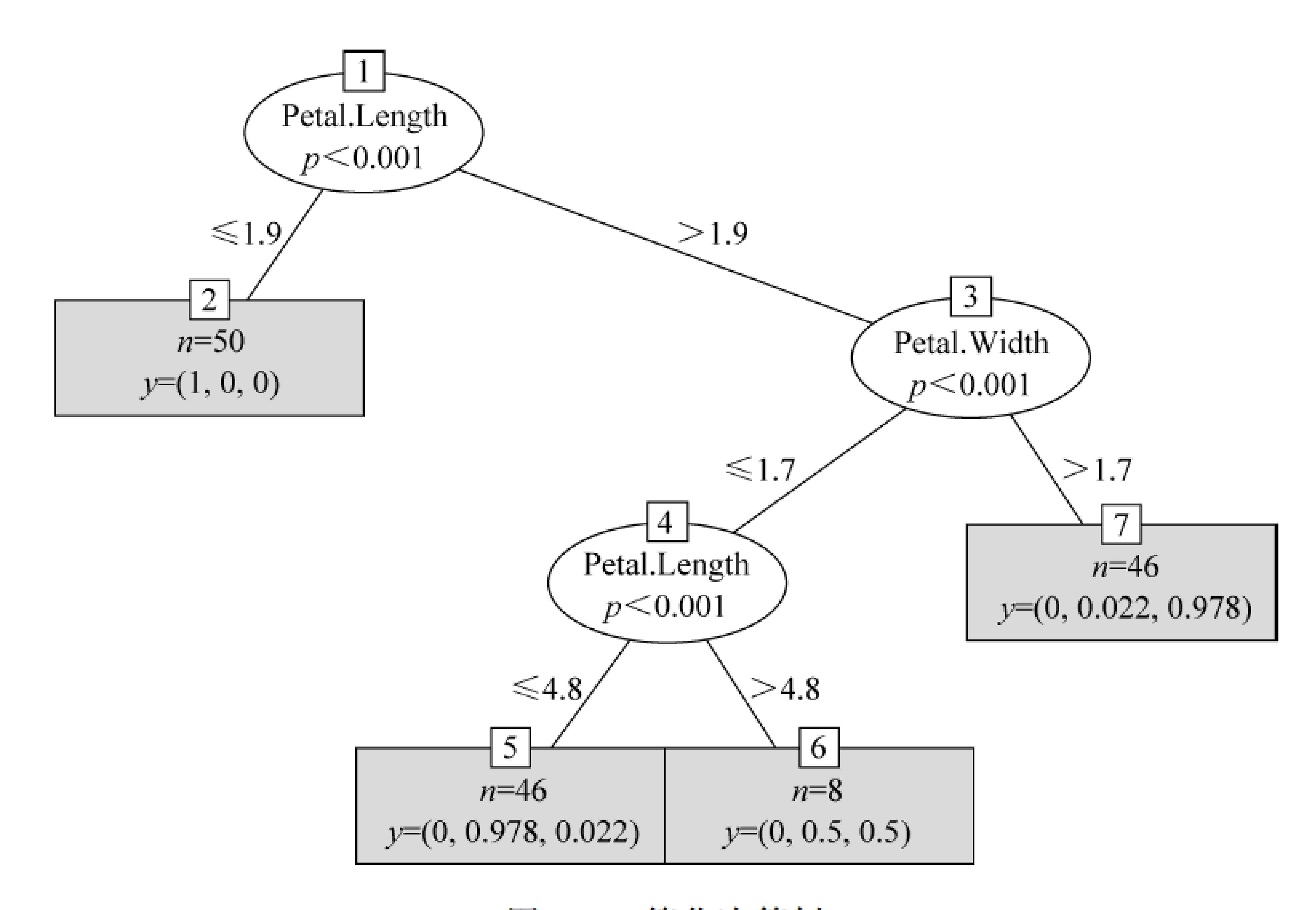


图 9.7　简化决策树

```
> plot(iris_ctree, type = "simple")
```

9.6 人工神经网络

9.6.1 三层前馈神经网络原理

人工神经网络(Artificial Neural Networks,ANNs)是一种模仿动物神经网络行为特征,进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度,通过调整内部大量节点之间相互连接的关系,达到处理信息的目的。人工神经网络具有自学习功能、联想存储功能和高速寻找优化解的能力,可以为人类提供经济预测、市场预测和效益预测,应用前途很远大。

最广泛应用于预测的神经网络是三层单向传播的前馈神经网络,由一个输入层、一个输出层和一个隐含层组成。信息由输入层进入网络,向前逐层传播至隐含层,再由输出层输出。前馈神经网络(Feedforward Networks)也称为多层感知器(Multilayer Perceptron,MLP)模型,可以通过增设隐含层结点数来一致逼近任何连续函数,神经网络的这个性质也被称为多层感知器的一般逼近性质。调整隐含层数、神经网络输入层和输出层的激活函数,可以形成不同形式的前馈神经网络结构。单个隐含层前馈神经网络的结构如图 9.8 所示。

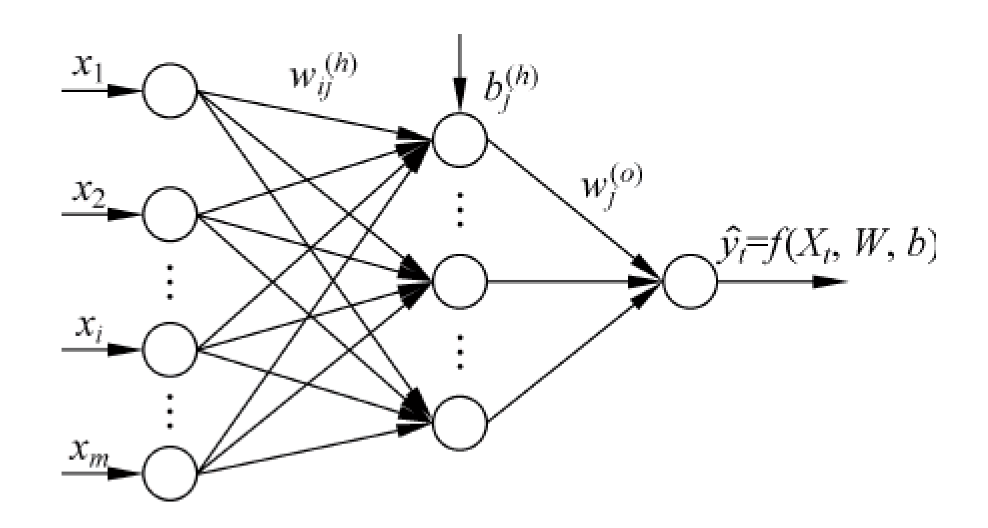


图 9.8　单个隐含层前馈神经网络结构模型

神经网络在隐含层和输出层都通过激活函数(Activation Function)来处理上一层向下一层传输的处理加工信息。选用不同的激活函数,可以使得神经网络结构表示成各种不同的数学模型。

在隐含层的第 j 个结点可以得到输出信息。

$$h_j = g_j^{(h)}\left(\sum_{i=1}^{m} w_{ij}^{(h)} x_i + b_j^{(h)}\right)$$

其中,$x_i(i=1,2,\cdots,m)$为输入层的第 i 个结点的输入变量;$w_{ij}^{(h)}(j=1,2,\cdots,n)$为输入层到隐含层的连接权重;$b_j^{(h)}$ 为隐含层阈值;$g_j^{(h)}(\cdot)$为隐含层激活函数,一般采用 sigmoidal 函数。

联合各层输入信息,可以在输出层得到输出信息。

$$f(\boldsymbol{X}_t,\boldsymbol{W},\boldsymbol{b}) = g_j^{(o)}\left\{\sum_{j=1}^{n} w_j^{(o)} h_j(t) + b^{(o)}\right\}$$

式中,$w_j^{(o)}$ 为隐含层到输出层的连接权重。$b^{(o)}$ 为输出层阈值。$g_j^{(o)}(\cdot)$为输出层激活

函数，输出层的激活函数根据应用的不同而异，如果用于函数逼近，一般采用线性函数形式；如果用于分类，则选用阈值函数。

9.6.2 神经网络分类实例

分类是三层前馈神经网络的一种重要应用，下面利用 R 中的神经网络包 RSNNS 对鸢尾花(iris)数据集进行分类。运行该例子前需安装 RSNNS 包。

(1) 载入程序和数据。

```
> library(RSNNS)
> data(iris)
```

(2) 定义网络输入。

```
> irisValues = iris[,1:4]
```

(3) 定义网络输出，并将数据进行格式转换。

```
> irisTargets = decodeClassLabels(iris[,5])
```

从中划分出训练样本和检验样本，默认 15%划分为测试样本，此处划分后 18%的数据为测试数据。

```
> iris = splitForTrainingAndTest(irisValues, irisTargets, ratio = 0.18)
```

(4) 对数据进行标准化。

```
> iris = normTrainingAndTestSet(iris)
```

(5) 利用 mlp 命令执行前馈反向传播神经网络算法。

```
> model = mlp(iris $ inputsTrain, iris $ targetsTrain,
            size = 3, learnFunc = "Quickprop",
            learnFuncParams = c(0.1, 2.0, 0.0001, 0.1),
            maxit = 100, inputsTest = iris $ inputsTest,
            targetsTest = iris $ targetsTest)
Class: mlp -> rsnns
Number of inputs: 4
Number of outputs: 3
Maximal iterations: 100
Initialization function: Randomize_Weights
Initialization function parameters: - 0.3 0.3
Learning function: Quickprop
```

```
Learning function parameters: 0.1 2 1e-04 0.1
Update function:Topological_Order
Update function parameters: 0
Patterns are shuffled internally: TRUE
Compute error in every iteration: TRUE
Architecture Parameters:
 $ size
[1] 3

All members of model:
[1] "nInputs"                  "maxit"              "initFunc"
[4] "initFuncParams"           "learnFunc"          "learnFuncParams"
[7] "updateFunc"               "updateFuncParams"   "shufflePatterns"
[10] "computeIterativeError"   "snnsObject"         "archParams"
[13] "IterativeFitError"       "IterativeTestError" "fitted.values"
[16] "fittedTestValues"        "nOutputs"
```

（6）利用上面建立的模型进行预测。

```
> predictions = predict(model,iris $ inputsTest)
```

（7）生成混淆矩阵，观察预测精度。

```
> confusionMatrix(iris $ targetsTest,predictions)
         predictions
  targets  2  3
  3        3  24
```

从上面混淆矩阵可以看出：行之和 27 为测试样本数据个数，24 表示第三类数据(Species 中的 virginica)有 24 个预测正确，另有 3 个预测错误归类为第二类数据(Species 中的 versicolor)。

上面的实例是对排好序的数据运用神经网络算法进行预测。当然，可以将原鸢尾花(iris)数据集的顺序打乱，进行数据乱序的预测，那么需要在上述步骤(1)、(2)之间加入以下代码：

```
> iris = iris[sample(1:nrow(iris),length(1:nrow(iris))),1:ncol(iris)]   #将数据顺序打乱
```

此时训练的模型信息如下：

```
Class: mlp->rsnns
Number of inputs: 4
Number of outputs: 3
Maximal iterations: 100
Initialization function: Randomize_Weights
```

```
Initialization function parameters: -0.3 0.3
Learning function: Quickprop
Learning function parameters: 0.1 2 1e-04 0.1
Update function:Topological_Order
Update function parameters: 0
Patterns are shuffled internally: TRUE
Compute error in every iteration: TRUE
Architecture Parameters:
 $size
[1] 3
All members of model:
[1] "nInputs"            "maxit"
[3] "initFunc"           "initFuncParams"
[5] "learnFunc"          "learnFuncParams"
[7] "updateFunc"         "updateFuncParams"
[9] "shufflePatterns"    "computeIterativeError"
[11] "snnsObject"        "archParams"
[13] "IterativeFitError" "IterativeTestError"
[15] "fitted.values"     "fittedTestValues"
[17] "nOutputs"
```

可得最后分类结果如下：

```
predictions
targets  1  2  3
      1 10  0  0
      2  0  6  2
      3  0  0  9
```

由混淆矩阵可以看出：各行之和 27 为测试样本数据个数。第一行表示第一类数据(Species 中的 setosa)有 10 个预测正确，没有错误划分；第二行表示第二类数据(Species 中的 versicolor)有 6 个预测正确，2 个错划分为第三类数据；第三行表示第三类数据(Species 中的 virginica)有 9 个预测正确，没有错误预测划分。

9.7 支持向量机

9.7.1 支持向量机原理

SVM(Support Vector Machine，支持向量机)法是建立在统计学习理论基础上的机器学习方法，由 Vapnik 等人于 1995 年提出，具有相对优良的性能指标。通过学习算法，SVM 可以自动寻找出那些对分类有较好区分能力的支持向量，由此构造出的分类器可以最大化类与类的间隔，因而有较好的适应能力和较高的分准率。该方法的特点是只由各类域的边界样本(支持向量)的类别来决定最后的分类结果，因此被称为支持向量机。

对于线性可分的数据，支持向量机算法的目的在于寻找一个超平面 $H(d)$，该超平面可

以将训练集中的数据分开，且与类的距离最大，故 SVM 法也被称为最大边缘（Maximum Margin）算法。待分样本集中的大部分样本不是支持向量，移去或者减少这些样本对分类结果没有影响。SVM 法对小样本情况下的自动分类有着较好的分类结果。从本质上看。SVM 避免了从归纳到演绎的传统过程，实现了高效的从训练样本到预测样本的“转导推理”，大大简化了通常的分类和回归问题。

对于线性不可分的数据，SVM 的方法是把样本“升维”，即向高维空间做映射，甚至是向无穷维空间做映射。图 9.9 是一个升维的图例。

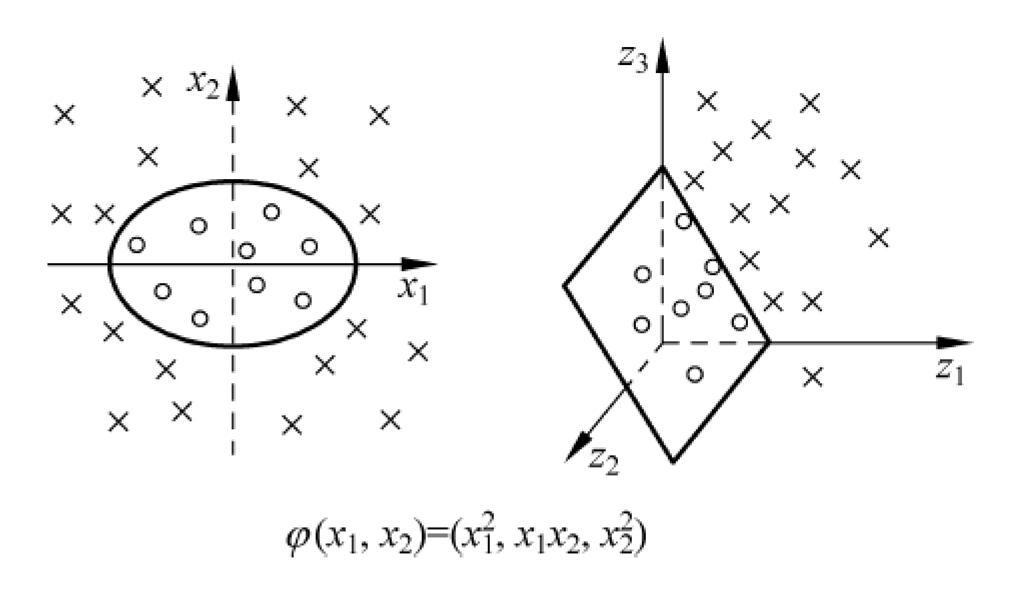


图 9.9　升维的图例

升维后再在高维空间中采用线性问题的方法。SVM 通过核函数实现到高维空间的非线性映射，从而可以解决样本空间中的高度非线性问题。图 9.10 清晰地展示了非线性映射的概念。

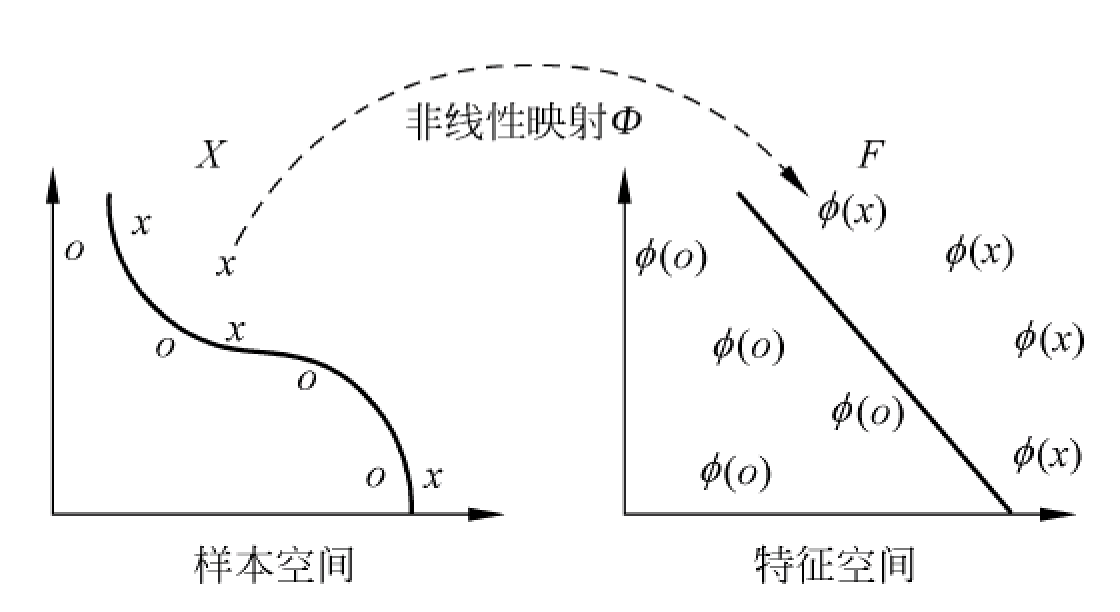


图 9.10　非线性映射的图例

9.7.2　支持向量机分类实例

下面在 R 环境下运用 svm 对 iris 数据集进行分类。运行该例子需要先安装 e1071 这个包。

（1）包的加载。

```
> library(e1071)
```

（2）数据集。

```
> data(iris)
```

(3) svm。

同样可以将数据集进行划分，也取后 27 个数据为测试数据。

```
> x <- subset(iris, select = -Species)
> y <- Species
> xtrain <- x[1:123,]
> ytrain <- y[1:123]
> xtest <- x[124:150,]
> ytest <- y[124:150]
> model <- svm(xtrain, ytrain,decision.values = TRUE, probability = TRUE)
> pred <- predict(model, xtest)
> summary(model)
Call:
svm.default(x = xtrain, y = ytrain, probability = TRUE, decision.values = TRUE)
Parameters:
   SVM-Type: C-classification
 SVM-Kernel: radial
       cost: 1
      gamma: 0.25
Number of Support Vectors: 41
(8 18 15)
Number of Classes: 3
Levels:
setosa versicolor virginica
```

(4) 将样本外数据应用于模型进行预测。

```
> pred <- predict(model, xtest)
```

(5) 以 table 形式输出预测结果。

```
> table(pred, ytest)
            ytest
pred          setosa versicolor virginica
  setosa        0        0          0
  versicolor    0        0          4
  virginica     0        0          23
```

从上述 table 可以看到，在 27 个测试样本预测中，第三类数据 virginica 有 23 个预测正确，有 4 个预测错误，误归类为第二类数据 versicolor。

同样，也可以将数据顺序打乱，进行乱序预测。在(2)、(3)步之间加入如下代码：

```
> iris = iris[sample(1:nrow(iris),length(1:nrow(iris))),1:ncol(iris)]
```

此时训练的模型如下：

```
Call:
svm.default(x = xtrain, y = ytrain, probability = TRUE, decision.values = TRUE)
Parameters:
   SVM - Type: C - classification
 SVM - Kernel: radial
       cost: 1
      gamma: 0.25
Number of Support Vectors: 45
( 17 9 19 )
Number of Classes: 3
Levels:
 setosa versicolor virginica
```

可以得出最终预测结果：

```
            ytest
pred          setosa versicolor virginica
  setosa        12        0         0
  versicolor    0         8         1
  virginica     0         0         6
```

从上述 table 可以看到，在 27 个测试样本预测中，第一行数据表示第一类数据 12 个全部划分正确；第二行数据表示第二类数据有 8 个划分正确，1 个被错误划分为第三类数据；第三行数据表示第三类数据有 6 个且全部划分正确。

对比与神经网络的划分效果，可以看出，利用神经网络和支持向量机在对数据集进行分类预测时都达到了很好的预测效果，且预测效果相当。

CHAPTER 10

第10章 信息可视化专题

信息可视化研究的是信息资源的可视化呈现,有利于人们对数据信息的理解和分析。本章介绍 R 的可视化功能,展示运用 R 进行世界地图、中国地图及公路线图的绘制,并给出一个城市暴力犯罪分布的可视化实例。

10.1 绘制地图

R 语言提供了强大的可视化绘图函数,本节详细地介绍地图的绘制。表 10.1 和表 10.2 分别列出了常用的可视化绘图函数及可视化绘图概念的介绍。

表 10.1 可视化绘图部分函数

函　　数	功　　能
get_map()	获取地图函数,可基于位置名称和经纬度获取地图
ggmap()	主要画图函数,可对比参考 ggplot
qmap()	快速画图,整合 get_map+ggmap

表 10.2 可视化绘图概念介绍

概　　念	功　　能
映射(Mapping)	将数据中的变量映射到图形属性,映射控制二者之间的关系
标度(Scale)	标度负责控制映射后图形属性的显示方式
几何对象(Geom)	代表在图中实际看到的图形元素,如点、线、多边形等
统计变换(Stat)	对原始数据进行某种计算,如对二元散点图加上一条回归线
坐标系统(Coord)	坐标系统控制坐标轴并影响所有的图形元素,坐标轴可以进行变换以满足不同的需要
图层(Layer)	数据、映射、几何对象、统计变换等构成一个图层,图层允许用户一步步构建图形,方便单独对图层进行修改
分面(Facet)	条件绘图,将数据按照某种方式分组,然后分别绘图,分面就是控制分组绘图的方法和排列形式

10.1.1　世界地图

首先安装 maps 包(install.packages("maps")),这个包包含世界地图和美国地图的数据如图 10.1 所示。绘制世界地图的 R 代码如下①:

```
> library(maps)
> map("world", fill = TRUE, col = rainbow(200),ylim = c( - 60, 90), mar = c(0, 0, 0, 0))
title("世界地图")
```



图 10.1　世界地图

maps 包没有中国地图的数据。但在 mapdata 包中存有中国地图的数据,但是这些数据很久没有更新了。

10.1.2　中国地图

本节主要介绍在 R 环境下用以下两种方式绘制中国地图。

(1) mapdata 包中读取中国地图数据②如图 10.2 所示。

```
> library(maps)
> library(mapdata)
> map("china", col = "red4", ylim = c(18, 54), panel.first = grid())
title("中国地图")
```

(2) 从 Google 获取中国地图数据③,如图 10.3 所示。

```
> library(ggmap)
> library(mapproj)
> map <- get_map(location = 'China', zoom = 4)
> ggmap(map)
```

① http://cos.name/2013/01/drawing-map-in-r-era/.

② http://cos.name/2013/01/drawing-map-in-r-era/.

③ http://cos.name/2013/01/drawing-map-in-r-era/.



图 10.2　中国地图

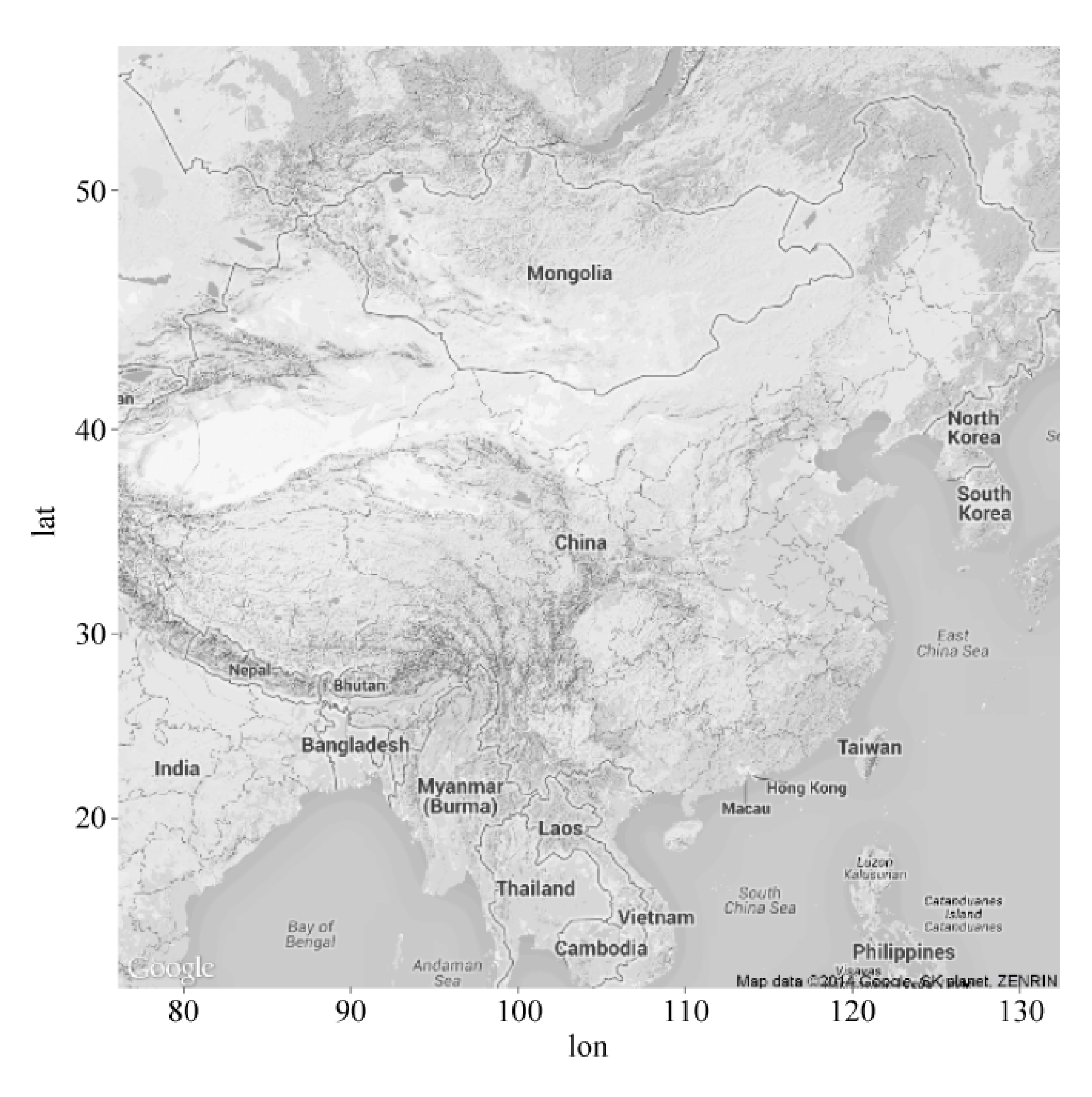


图 10.3　从谷歌获取的中国地图数据

10.1.3　公路线图

本节主要介绍在 R 环境下绘制广东省和深圳大学的公路线图。

1. 广东省的公路线图

从 Google 上获取广东省的公路地图数据，如图 10.4 所示。

```
> map <- get_map(location = 'Guangdong', zoom = 10, maptype = 'roadmap')
> ggmap(map)
```

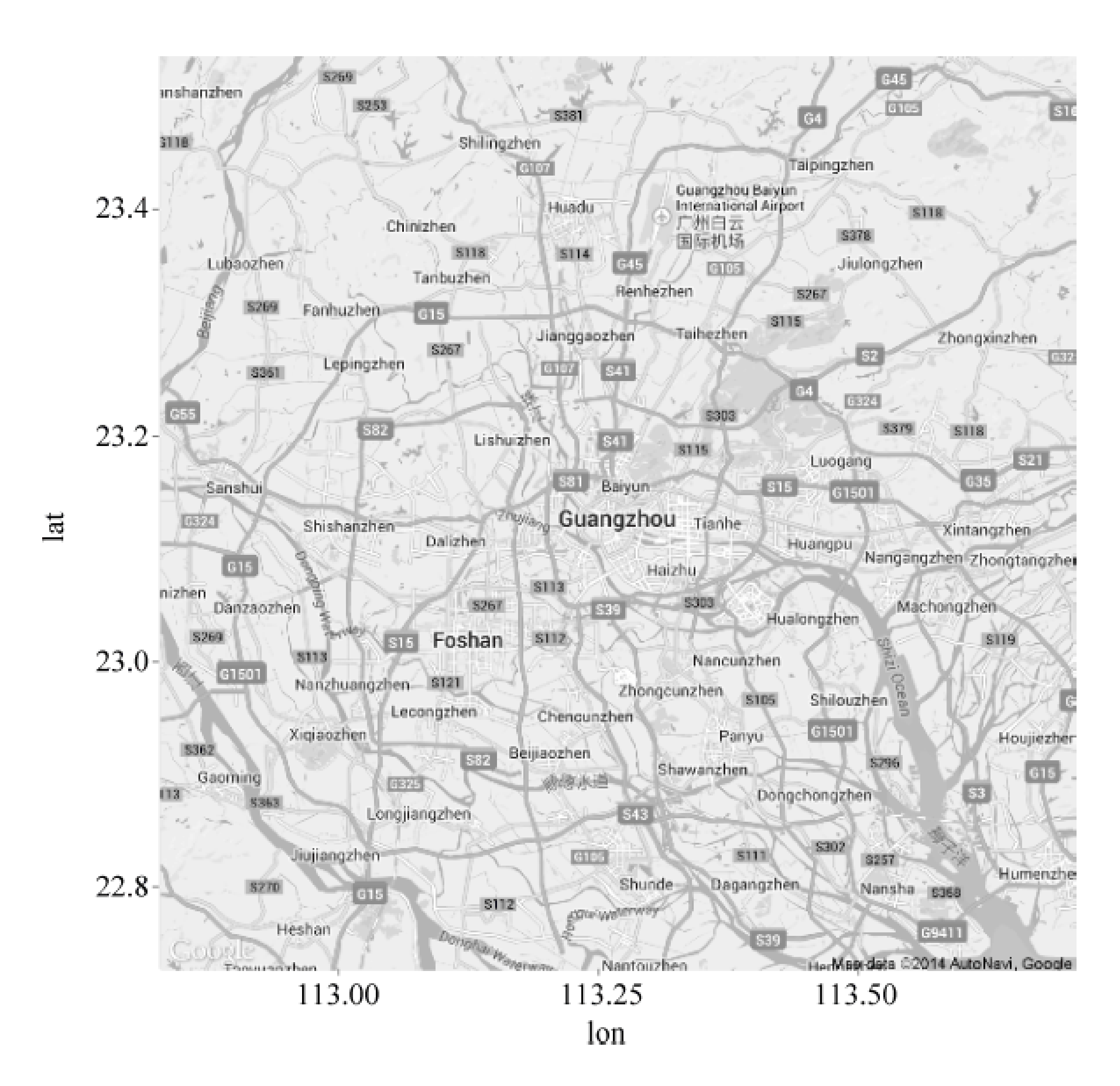


图 10.4　广东省公路线图

2. 深圳大学公路线图

从 Google 上获取深圳大学的公路地图数据，如图 10.5 所示。

```
> map <- get_map(location = 'Shenzhen University', zoom = 14,color = c("bw"), maptype = '
roadmap')
> ggmap(map)
```



图 10.5　深圳大学公路线图

10.2 可视化实例

10.2.1 数据

1. 数据集 Crime

ggmap 包中自带数据集 Crime,查看数据结构。

```
>library(ggmap) # 加载包
>str(crime) # 查看数据结构
'data.frame'  :86314 obs. of 17 variables:
 $ time      : POSIXt, format: "2010-01-01 14:00:00" "2010-01-01 14:00:00" ...
 $ date      : chr "1/1/2010" "1/1/2010" "1/1/2010" "1/1/2010" ...
 $ hour      : int 0 0 0 0 0 0 0 0 0 0 ...
 $ premise   : chr "18A" "13R" "20R" "20R" ...
 $ offense   : Factor w/ 7 levels "aggravated assault",..: 4 6 1 1 1 3 3 3 3 3 ...
 $ beat      : chr "15E30" "13D10" "16E20" "2A30" ...
 $ block     : chr "9600-9699" "4700-4799" "5000-5099" "1000-1099" ...
 $ street    : chr "marlive" "telephone" "wickview" "ashland" ...
 $ type      : chr "ln" "rd" "ln" "st" ...
 $ suffix    : chr "-" "-" "-" "-" ...
 $ number    : int 1 1 1 1 1 1 1 1 1 1 ...
 $ month     : Ord.factor w/ 8 levels "january"<"february"<..: 1 1 1 1 1 1 1 1 1 1 ...
 $ day       : Ord.factor w/ 7 levels "monday"<"tuesday"<..: 5 5 5 5 5 5 5 5 5 5 ...
 $ location  : chr "apartment parking lot" "road / street / sidewalk" "residence / house" "
residence / house" ...
 $ address   : chr "9650 marlive ln" "4750 telephone rd" "5050 wickview ln" "1050 ashland st"
...
 $ lon       : num -95.4 -95.3 -95.5 -95.4 -95.4 ...
 $ lat       : num 29.7 29.7 29.6 29.8 29.7 ...
```

2. 找到一个合理的空间范围

由于主要关注的是某个城市暴力犯罪发生的分布情况,因此可以对数据设置此约束条件。为了确定边界框,首先使用 gglocator()返回经度和纬度的坐标值(图 10.6)。

```
>qmap('houston', zoom = 13)
```

```
>gglocator(2)#返回经度和纬度的坐标值
       lon        lat
1  -95.39487  29.77949
2  -95.39068  29.77342
```



图 10.6　获取 houston 地图数据

3. 返回符合条件的子集

考虑严重袭击、抢劫、强奸、谋杀 4 类犯罪,筛选符合条件的数据返回子集。

```
# only violent crimes
violent_crimes <- subset(crime,offense != "auto theft" &
                    offense != "theft" & offense != "burglary")
# order violent crimes
violent_crimes$offense <- factor(
       violent_crimes$offense, levels =
       c("robbery", "aggravated assault",
       "rape", "murder"))
# restrict to downtown
violent_crimes <- subset(violent_crimes,
                          -95.39681 <= lon & lon <= -95.34188 &
                           29.73631 <= lat & lat <= 29.78400)
```

10.2.2　ggmap

(1) 查看个人犯罪在哪些地方发生,生成空间气泡分布图,如图 10.7 所示。

```
>theme_set(theme_bw(16))
>HoustonMap <- qmap("houston", zoom = 14,color = "bw", legend = "topleft")
HoustonMap +
  geom_point(aes(x = lon, y = lat,
                 colour = offense, size = offense),
         data = violent_crimes)
```

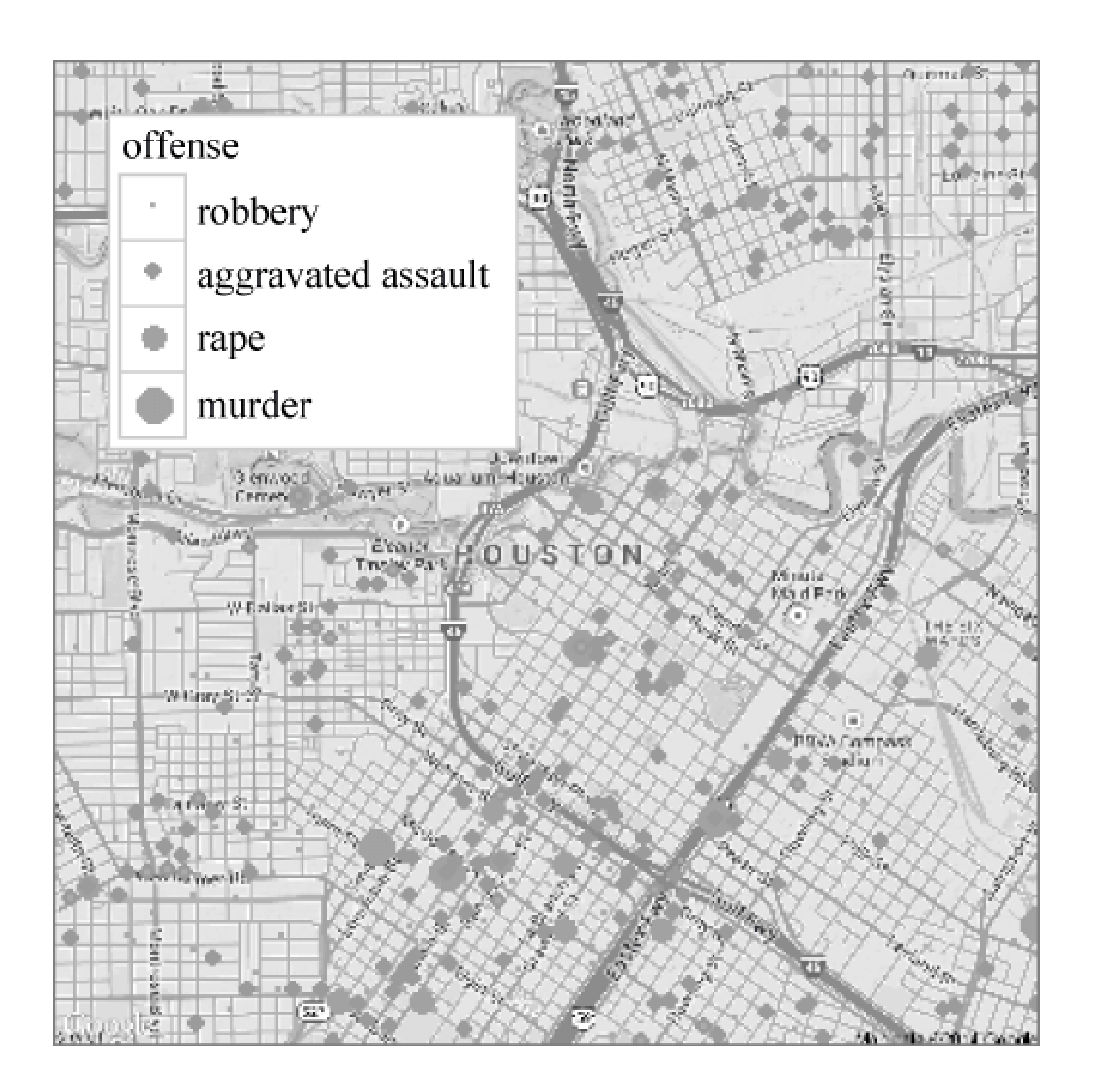


图 10.7 犯罪分布空间气泡图

（2）从图 10.7 所示的气泡图可以发现一个问题：不能直观地从图中感受到哪种暴力犯罪发生和其分布情况。因此，可以通过 stat_bin2d()绘制下列分布图，从而清晰地看出犯罪发生的区域情况和频率，如图 10.8 所示。

```
HoustonMap +
  stat_bin2d(aes(x = lon, y = lat, colour = offense,
        fill = offense),
    size = .5, bins = 30, alpha = 1/2,
    data = violent_crimes)
```

（3）由图 10.8 可以看出 ggplot2 绘图的强大功能。如果忽略 offense 类型，可以使用等高线图绘制犯罪空间分布图，也可以获得很好的效果。graphics 包里的 filled.contour()也可以产生一个纯色填充区域的等高线图，如图 10.9 所示。

```
> houston <- get_map("houston", zoom = 14)
> HoustonMap <- ggmap("houston",extent = "device", legend = "topleft")
HoustonMap +
  stat_density2d(
    aes(x = lon, y = lat, fill = ..level..,
        alpha = ..level..),
    size = 2, bins = 4, data = violent_crimes,
    geom = "polygon")
```

（4）图 10.9 的重叠可以有非常好的效果，但是它所展示的信息也有可能被重叠部分隐藏，特别是在使用有颜色的地图时更加常见。为此，可以利用 inset()在地图上插入一个白色背景的重叠图样。

```
> overlay <- stat_density2d(
  aes(x = lon, y = lat, fill = ..level..,
      alpha = ..level..),
  bins = 4, geom = "polygon",
  data = violent_crimes)
HoustonMap + overlay + inset(
  grob = ggplotGrob(ggplot() +
                    overlay + theme_inset()),
  xmin = -95.35836, xmax = Inf,
  ymin = -Inf, ymax = 29.75062
)
```

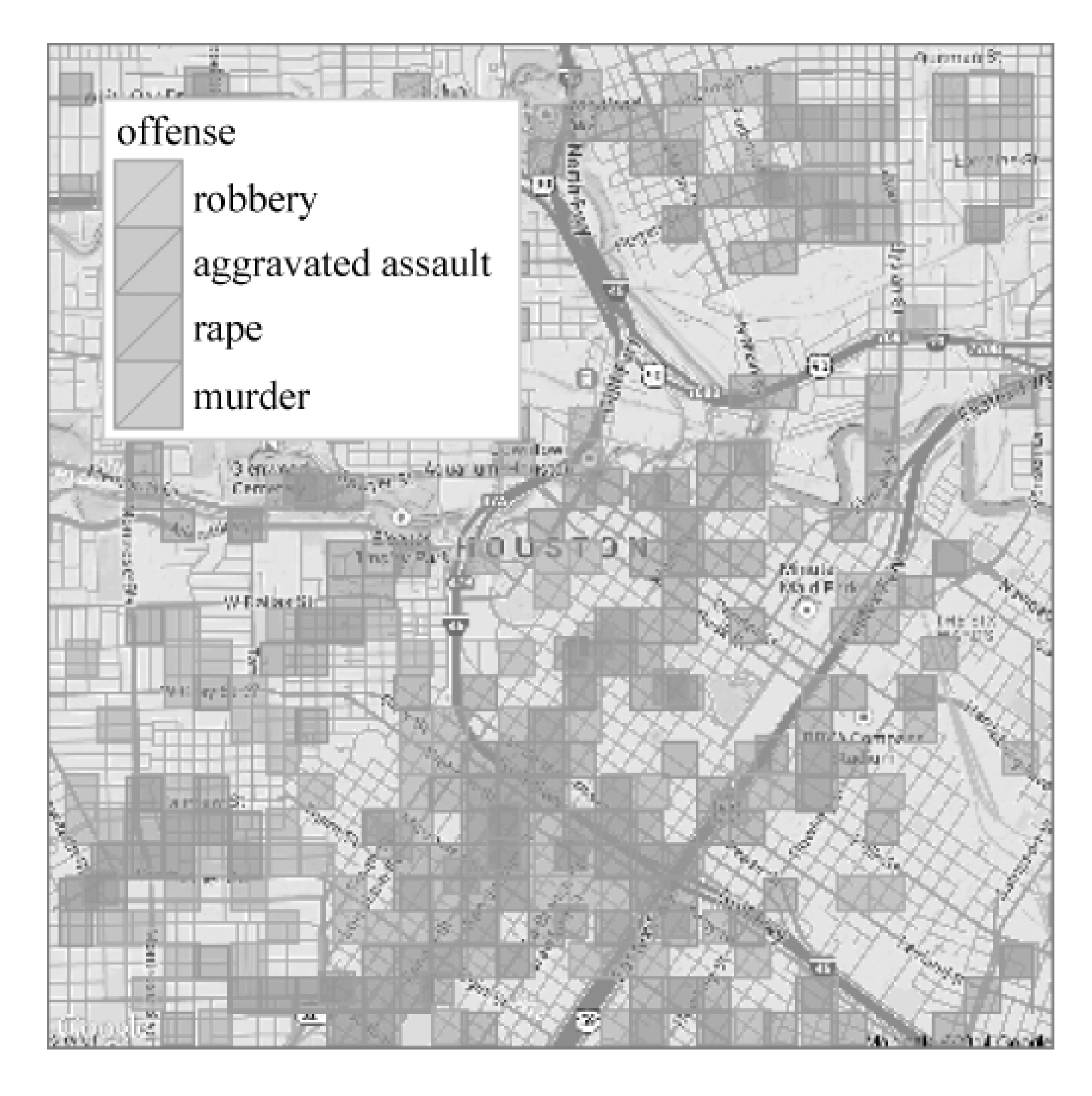


图 10.8 犯罪分布 stat_bin2d()图

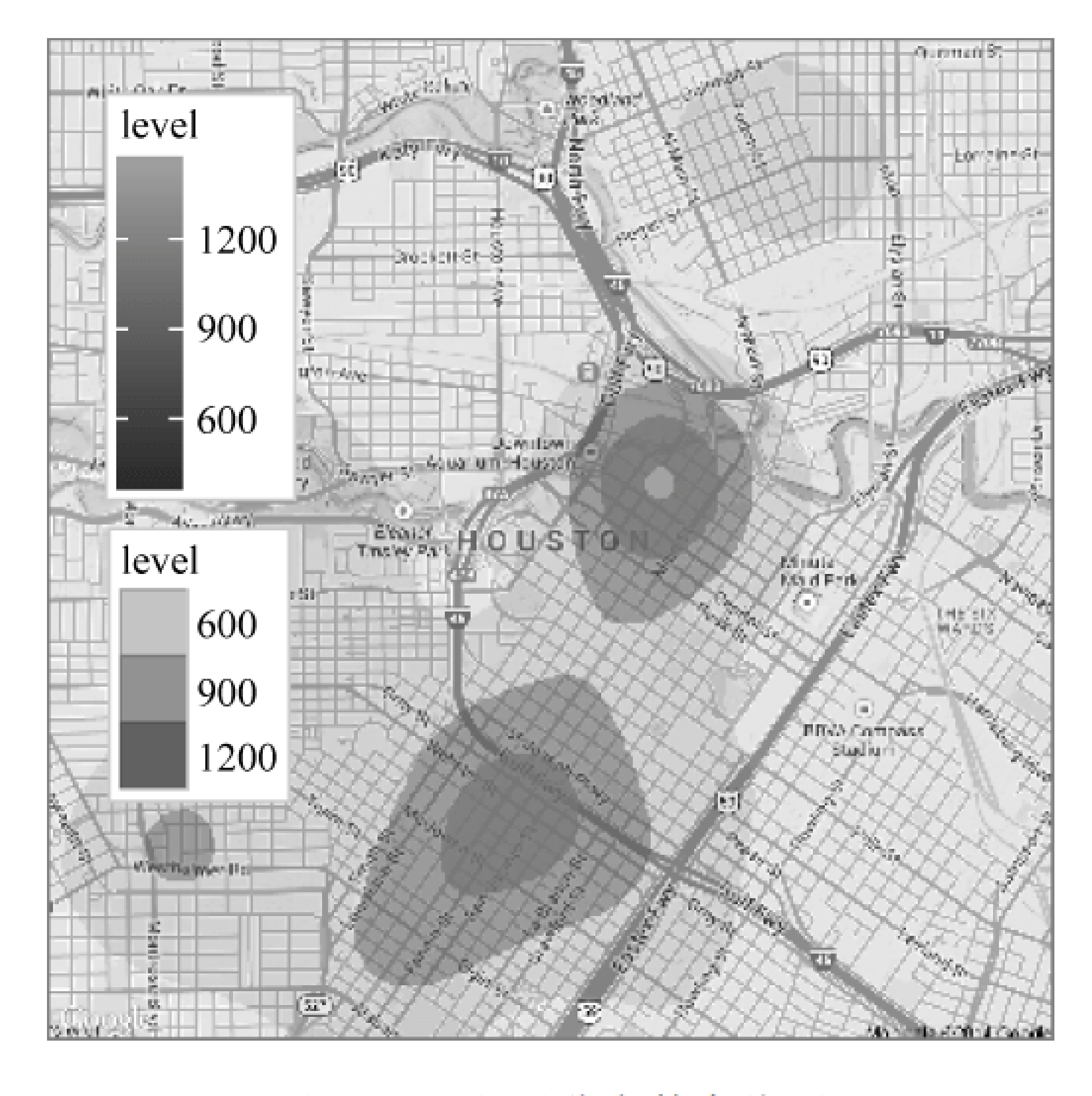


图 10.9 犯罪分布等高线图

由图 10.10 可以看出犯罪高频发生的主要三大地区。这三个对应的区域确实是 Houston 经常发生犯案的地方。由东往西，犯案的主要原因是由于监狱里释放的犯人在此游荡，商业总站有许多贫困、无家可归的人，流动人口较大的地方。

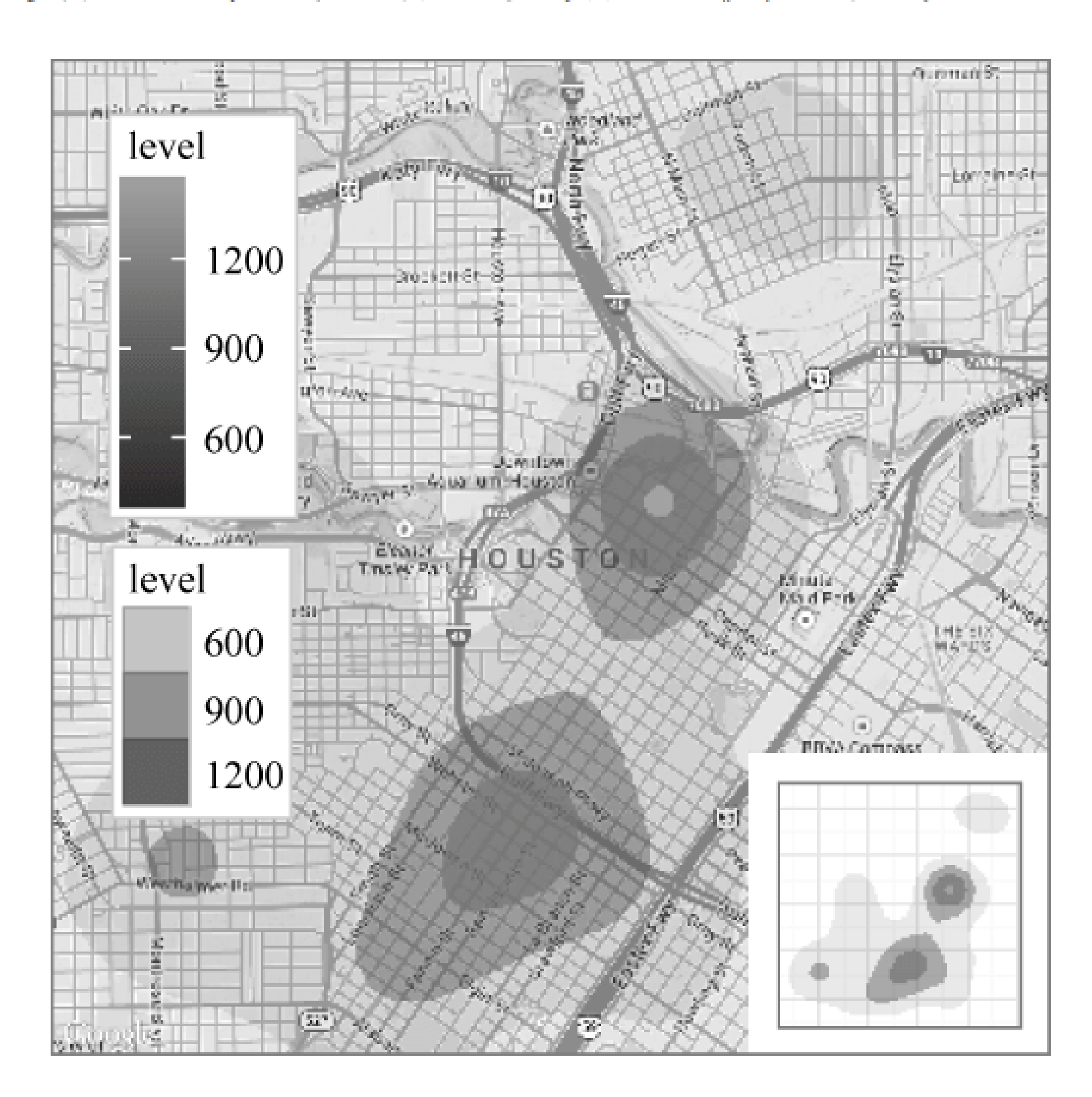


图 10.10 犯罪分布等高线图

(5) 除了单方位绘图外，还可以从多方位去绘制，如可以通过设置 ggmap() 和 qmap() 中的参数 base_layer 实现。这在时空数据中应用非常广泛，例如将时间分量分解为天、月、季、年等。

```
> houston <- get_map(location = "houston",
                     zoom = 14, color = "bw", source = "osm")
> HoustonMap <- ggmap(houston,
                      base_layer = ggplot(aes(x = lon, y = lat),
                                          data = violent_crimes))
HoustonMap +
  stat_density2d(aes(x = lon, y = lat,fill = ..level.., alpha = ..level..),
                 bins = 5, geom = "polygon",
                 data = violent_crimes) +
  scale_fill_gradient(low = "black",
                      high= "red") +
  facet_wrap(~ day)
```

图 10.11 展示了从不同角度分解等高线图。从图中可以看出，星期一犯罪发生频率最高，星期五排第二。城市中心酒吧区和周边区活跃的夜生活可以解释星期五的犯罪发生频率高的现象。星期一犯罪频率高也许可以通过监狱周一释放犯人最多来解释。

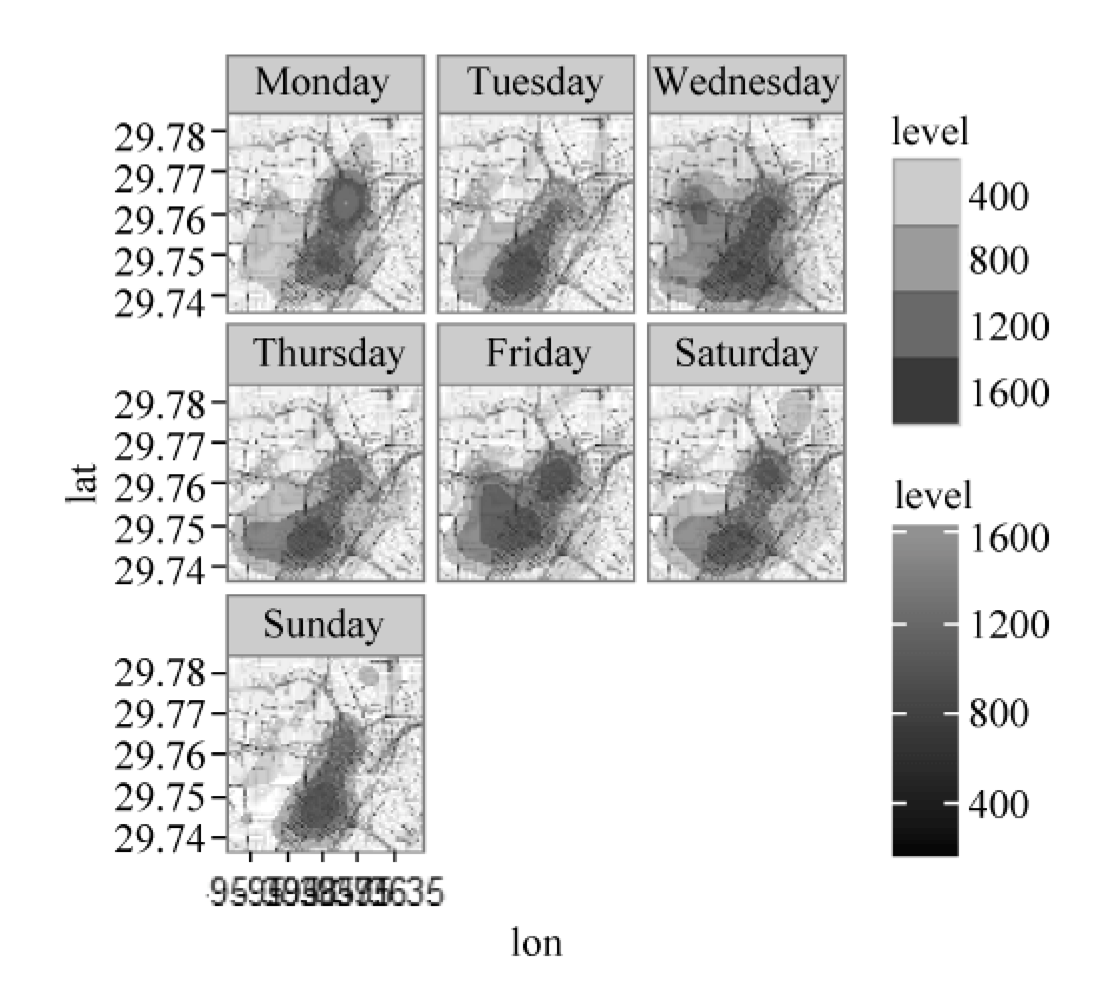


图 10.11　犯罪分布分面图

PART 4

第四部分

RHadoop案例分析

CHAPTER 11

第11章

RHadoop的基本操作

RHadoop结合了R与Hadoop两方面的优势,使得数据的处理分析变得更加完美。

一方面,R是一个可以对数据进行统计分析的开源软件包,有着非常强大的软件包基础,各个学界的精英人士会在R的平台上编写相关的程序包,并且程序开源,可以供大众进一步完善并使用。对于初学者来说,R提供的数据分析平台无论从数据读取,还是方法的应用都十分简便。但R的可拓展性较差,R的核心技术引擎只能加工和处理有限的数据量,一旦数据量较大,R的运行速度会大大降低,甚至不能运行。

另一方面,Hadoop在处理大数据上十分流行,其优势在于能够存储处理TB甚至PB级的数据。那么,将R和Hadoop结合正好取长补短,发挥两者的优势,实现大规模数据的高效处理分析。

下面介绍一些在RHadoop平台上的基本操作。首先需要加载包以完成Rhadoop环境的准备工作。

(1) 确定工作路径。

```
>getwd()
```

(2) 加载相应的软件包,以及初始化hdfs。

```
>library(rhdfs)
>hdfs.init()
>library(rmr2)
```

11.1 数据文件的读取

在Rhadoop的环境下读取数据的原理与在R中读取数据一致。首先用getwd()命令获取R的工作路径,把数据文件存入工作目录中。然后运用R命令进行数据文件的读取,

例如读取 txt 文件使用命令 read.table()；读取 csv 文件使用命令 read.csv()等。下面以读取 goods.csv 数据文件为例。

（1）读取数据。

```
> data <- read.csv("goods.csv",header = T)
```

（2）把数据加载到 HDFS 中并读取。

```
> data.dfs <- to.dfs(keyval(1,data))   # 将 data 加载到 HDFS 中并赋值给 data.dfs
> from.dfs(data.dfs)   # 从 HDFS 中读取 data.dfs
Skey
[1]  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
Sval
       user_ID item_ID item_Reting
1         1     101     5.0
2         1     102     3.0
3         1     103     2.5
4         2     101     2.0
5         2     102     2.5
6         2     103     5.0
7         2     104     2.0
8         3     101     2.0
9         3     104     4.0
10        3     105     4.5
11        3     107     5.0
12        4     101     5.0
13        4     103     3.0
14        4     104     4.5
15        4     106     4.0
16        5     101     4.0
17        5     102     3.0
18        5     103     2.0
19        5     104     4.0
20        5     105     3.5
21        5     106     4.0
```

11.2 包的加载

RHadoop 环境中 R 包的安装和加载命令与 R 环境中的相同，下面以安装加载作图软件包 ggplot2 包为例，并简要说明 ggplot2 包在直方图和密度函数图的应用。使用 ggplot2 软件包加载一份钻石的数据，并对钻石的数据分别作出直方图（如图 11.1 所示）和密度函数图（如图 11.2 所示），操作命令如下：

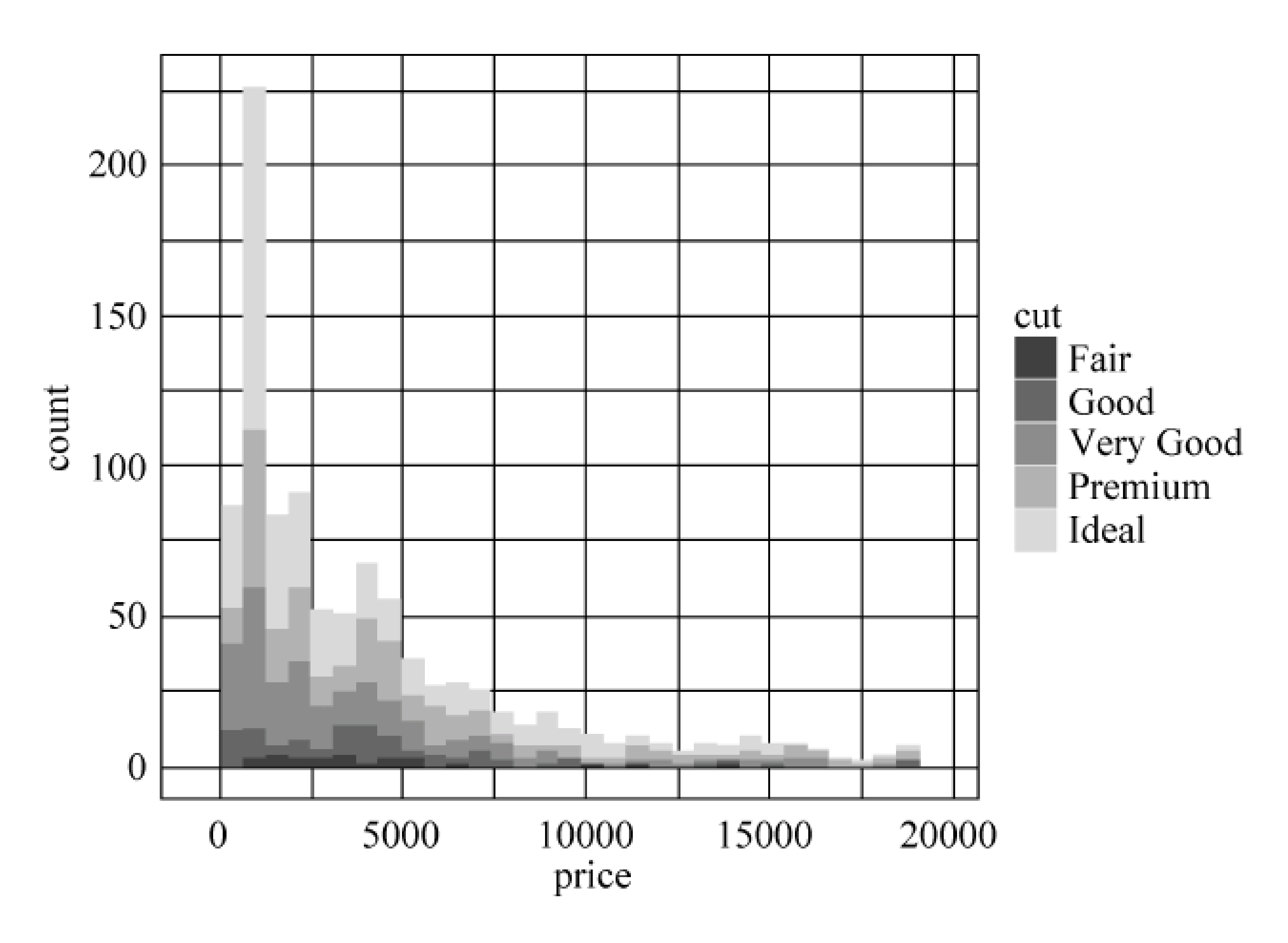


图 11.1　直方图

```
> install.packages("ggplot2")
> library(ggplot2)
> data(diamonds);set.seed(42)
> small <- diamonds[sample(nrow(diamonds), 1000), ]
> ggplot(small) + geom_histogram(aes(x = price, fill = cut))
```

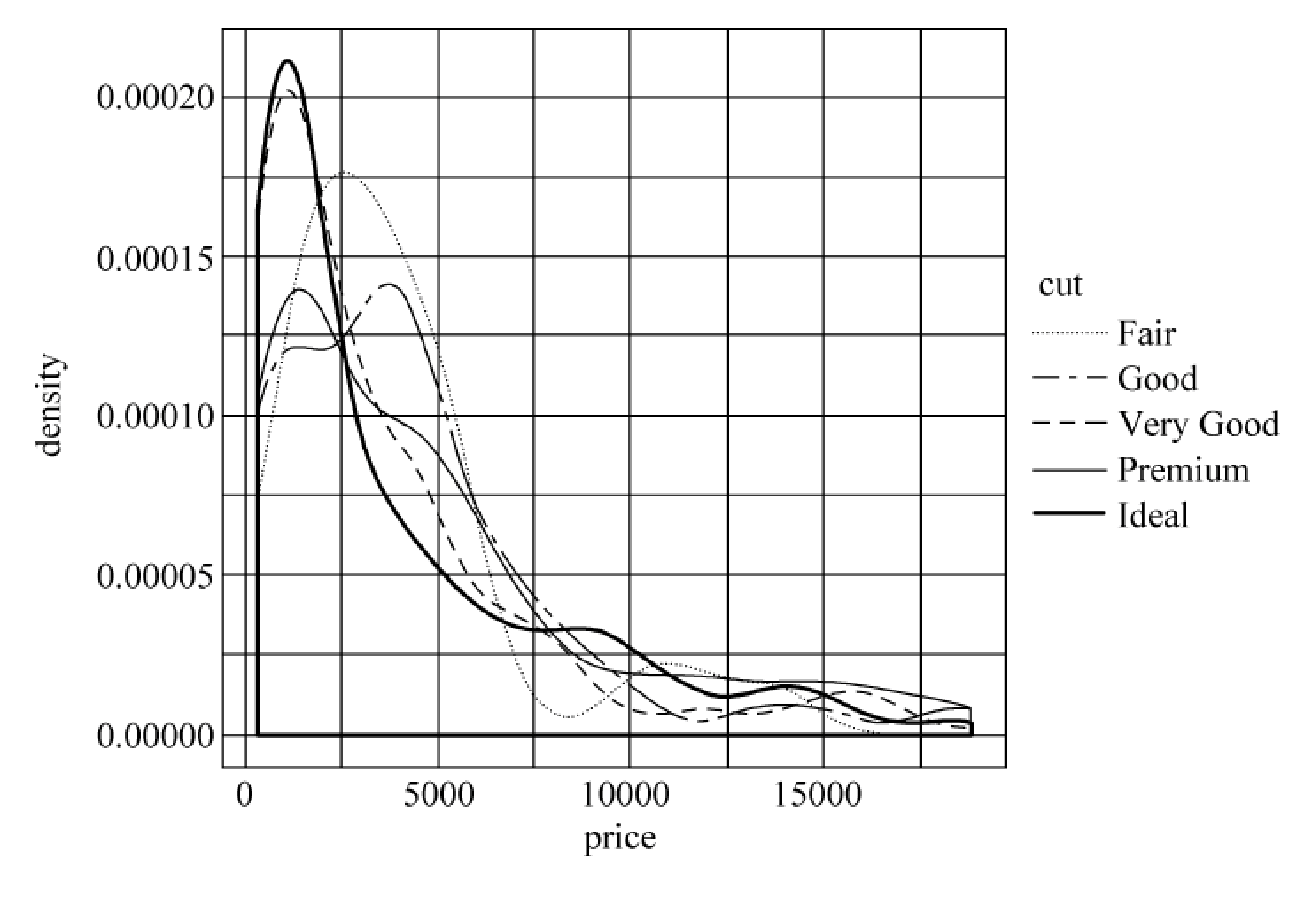


图 11.2　密度函数图

```
> ggplot(small) + geom_histogram(aes(x = price, fill = cut))
```

11.3　基本函数

在 RHadoop 环境中，R 软件的相关操作都可以运行。表 11.1 列举了一些 R 中的基本

操作函数，可以帮助初学者尽快了解 RHadoop。

表 11.1　R 软件的一些基本函数

函　　数	用　　法
getwd()	查看当前的工作目录
setwd()	设定当前的工作目录
help()/?	帮助函数
example()	展示函数的用法
help.search("")/??""	查找一个不太确定的内容
help(package=)	返回整个包的基本信息

在 RHadoop 中，数据的读取与 R 软件有一些区别，具体读取数据的函数如表 11.2 所示。

表 11.2　R 与 RHadoop 读取数据的函数比较

	函　　数	用　　法
R 软件	read.table()	读入文本数据
	read.csv()	读入 csv 格式文件
	odbcConnectExcel()	读取 Excel 数据文件
	sqlFetch()	读取 Excel 数据表单
	write.csv()	输出 csv 文件
RHadoop	to.dfs()	输入数据到 HDFS
	from.dfs()	从 HDFS 中导出数据
	keyval(key,val)	创建及提取键值对

在 RHadoop 中关键是要理解 MapReduce 的工作原理，Hadoop 不是一个对所有大数据问题的通用解决方案。它只是把预处理的大数据分割成小块，并通过分布式的服务实现并行处理而已，这使得在处理大数据时可以节省更多的时间将成本降低。

Hadoop 的数据处理进程包括多个任务，这些任务可以帮助在输入数据集中得到最终的输出数据，主要包括：

① 预载数据到 HDFS 上。

② 通过调用 Driver 来运行 MapReduce。

③ Map 输入数据的读取，该任务会把数据进行分割并执行 Map 内部的逻辑，最后生成中间阶段的键值对。

④ 执行合并（Combiner）和重组（Shuffle）阶段主要用于优化 Hadoop MapReduce 进程。

⑤ 排序（Sorting）并提供中间数据（键值对数据）给 Reduce 阶段作为输入，然后执行 Reduce 阶段的程序。Reduce 执行单元处理这些分割的键值对数据，并依据 Reduce 内的函数逻辑对该数据进行聚集（Aggregate）。

⑥ 把排序后的最终数据存储于 HDFS 文件系统中。

CHAPTER 12

第12章

RHadoop环境下案例分析

12.1 回归分析

弗朗西斯·高尔顿在1877年发表关于种子的研究结果,指出“回归到平均值”现象的存在。他曾对亲子间的身高做研究,发现父母的身高虽然会遗传给子女,但子女的身高却有逐渐“回归到中等(即人的平均值)”的现象,这个概念与现代统计学中的“回归”并不相同,但却是“回归”一词的起源。在此后的研究中,高尔顿第一次使用了相关系数的概念。

卡尔·皮尔逊继弗朗西斯·高尔顿之后发展了与回归相关的理论,得到母体的概念,并认为统计研究不是样本本身,而是根据样本对母体的推断。由此导出了拟合优度检验:作为样本取出的若干个体分布是否与母体分布一致。此外,他还提出了净相关、复相关、总相关、相关比等概念,提出了计算复相关和净相关的方法及相关系数的公式,对统计学理论和回归分析方法的发展作出了重要的贡献。

从最初的理论提出到现在,回归分析已经发展得非常成熟。在实际应用中,它是数理统计学与实际问题联系最为紧密,应用范围最广泛,收效最为显著的统计分析方法,也是分析数据,寻求变量之间关系的有力工具,在生物、医学、农业、林业、经济、管理、金融、社会等领域应用广泛。

12.1.1 回归分析原理

回归分析是研究被解释变量(因变量)与解释变量(自变量)之间相互依赖的定量关系的一种统计分析方法。按照涉及的解释变量的多少,可分为一元回归分析和多元回归分析;按照因变量和自变量之间的关系类型,可分为线性回归分析和非线性回归分析。如果在回归分析中只包括一个自变量和一个因变量,且二者的关系可用一条直线近似表示,这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量,且因变量和自变量之间是线性关系,则称为多元线性回归分析。

回归分析有如下基本假定：

① 解释变量是确定型变量。

② 随机干扰项 μ_i 符合零均值、同方差、无序列相关性，$E(\mu_i)=0$，$\text{var}(\mu_i)=\sigma^2$，$\text{cov}(\mu_i,\mu_j)=0(i\neq j)$。

③ 随机干扰项与解释变量之间不相关，即 $\text{cov}(\mu_i,\mu_j)=0(i\neq j)$。

④ 随机干扰项服从零均值、同方差、零协方差的正态分布，即 $\mu_j\sim N(0,\sigma^2)$。

⑤ 随着样本容量的无限增加，解释变量 X 的样本趋于一个有限的常数。

⑥ 回归模型是正确设定的。

现实生活中的很多数据不是严格符合这些基本假定的，通常的回归分析更多的是近似的理论分析。

线性回归是最基本的回归分析方法。所谓线性回归方程，实际上就是将一系列测量数据通过数学方法来处理以确定相应的直线方程：

$$y = ax + b$$

只要求解出直线方程的两个系数 a 和 b，即确立了拟合方程。通常求解拟合方程未知系数的方法有端点法、平均法、最小二乘法，其中最小二乘法所得拟合直线精度最高，平均法次之，端点法较差。这里仅介绍一下最小二乘法的求解公式，这也是应用最广泛的方法。

最小二乘法的出发点是使实际测量数据 y_i 与拟合直线 $y=a+bx$ 上对应的估计值 $\hat{y}_i$ 的残差的平方和为最小。即

$$\sum_{i=1}^{n}V_i^2 = \sum_{i=1}^{n}[y_i-(a+bx_i)]^2 = \min$$

为使 $\sum_{i=1}^{n}V_i^2$ 的值最小，只要使 a 和 b 的偏导数为 0 即可解得 a 和 b 的值。经过求导之后得到 a 和 b 的计算公式：

$$\begin{cases} a = \dfrac{\sum_{i=1}^{n}x_i\sum_{i=1}^{n}x_iy_i - \sum_{i=1}^{n}y_i\sum_{i=1}^{n}x_i^2}{\left(\sum_{i=1}^{n}x_i\right)^2 - n\sum_{i=1}^{n}x_i^2} \\ b = \dfrac{\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i - n\sum_{i=1}^{n}x_iy_i}{\left(\sum_{i=1}^{n}x_i\right)^2 - n\sum_{i=1}^{n}x_i^2} \end{cases}$$

12.1.2 线性回归分析案例

在数据量较少的情况下，线性回归分析是可以手动计算的，但是在数据量较多时，一般采用统计软件来求解。很多统计软件，如 SPSS、Eviews、stata、R 都有相关的程序来解决回归分析问题，但是随着社会的发展，现在步入了大数据时代，一般的统计软件对于数据量非常大的情况往往计算的效率较低。Rhadoop 是 R 软件和 Hadoop 平台的结合，既能很好地发挥 R 语言的特长，又可借助 Hadoop 平台在大数据领域大展拳脚。本文主要研究线性回归分析在 RHadoop 集群环境下的求解。

线性回归模型一般表示如下：

$$\hat{Y} = a + bx$$

其中，$\hat{Y}$是由自变量 X 推算应变量 Y 的估计值，a、b 为待估系数，b 也称为样本的回归系数，这个案例就是在 RHadoop 集群环境下对回归系数矩阵进行求解。具体代码如下：

（1）设置随机数的状态。

```
set.seed(1234)
```

（2）大数据集的生成。

```
X <- matrix(rnorm(200000),ncol = 10)
X. index  <- to.dfs(cbind(1:nrow(X),X))
Y <- as.matrix(rnorm(20000))
```

（3）定义 map 作业 1 中执行单元的函数。

```
mapper1 <- function(.,Xi){
  Xi <- Xi[, -1]
  keyval(1,list(t(Xi) % * % Xi))
}
```

（4）定义 map 作业 2 中执行单元的函数。

```
mapper2 <- function(.,Xi){
  Yi <- Y[Xi[,1],]
  Xi <- Xi[, -1]
  keyval(1,list(t(Xi) % * % Yi))
}
```

（5）定义 reduce 执行单元中使用的函数，用于计算 map 执行单元输出的总和。

```
Sum.reduce <- function(.,YY){
  keyval(1,list(Reduce('+',YY)))
}
```

（6）用 Mapreduce 作业 1 计算 Xt • X。

① 调用 mapreduce 计算框架，产生 Xt • X 的 mapreduce 作业。

```
MP1 <- mapreduce(input = X.index, map = mapper1,
reduce = Sum.reduce,combine = TRUE)
```

② 将计算结果从 hdfs 载入 R 环境中，并取出相应的数值。

```
XtX <- values(from.dfs(MP1))[[1]]
```

第(6)步运行的结果矩阵，整理后如下：

	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
[1]	20075.68	−115.57	−94.09	134.96	115.43	64.23	33.03	165.55	−67.41	183.15
[2]	−115.57	19986.5	−76.73	−220.89	9.79	−246.38	−131.03	−136.93	−199.92	−91.62
[3]	−94.09	−76.73	20039.59	14.77	27.83	−25.87	−93.37	45.84	−91.7	−18.26
[4]	134.96	−220.89	14.77	19918.6	40.74	72.02	−98.88	236.45	97.15	−52.71
[5]	115.43	9.79	27.83	40.74	19810.18	−73.75	−49.44	−40.27	53.47	−302.53
[6]	64.23	−246.38	−25.87	72.02	−73.75	19532.87	113.9	69.74	271.77	−96.7
[7]	33.03	−131.03	−93.37	−98.88	−49.44	113.9	19778.85	105.61	−86.11	177.97
[8]	165.55	−136.93	45.84	236.45	−40.27	69.74	105.61	20316.07	−31.1	107.76
[9]	−67.41	−199.92	−91.7	97.15	53.47	271.77	−86.11	−31.1	19595.15	77.68
[10]	183.15	−91.62	−18.26	−52.71	−302.53	−96.7	177.97	107.76	77.68	20109.66

(7) 用 Mapreduce 作业 2 计算 Xt · Y。

```
MP2 <- mapreduce(input = X.index, map = mapper2,
reduce = Sum.reduce,combine = TRUE)
XtY <- values(from.dfs(MP2))[[1]]
```

同理，第(7)步运行的结果矩阵，整理后如下：

$$\boldsymbol{X}_t\boldsymbol{X}^{\mathrm{T}} = \begin{pmatrix} -40.62 \\ -123.06 \\ 26.97 \\ -103.15 \\ 58.28 \\ -210.08 \\ 129.58 \\ -122.07 \\ -199.37 \\ 325.01 \end{pmatrix}$$

(8) 计算回归的系数值。

```
solve(XtX,XtY)
```

最终整理后得到系数矩阵如下：

$$\boldsymbol{b} = \begin{pmatrix} -0.0021 \\ -0.0064 \\ 0.0013 \\ -0.0050 \\ 0.0032 \\ -0.0106 \\ 0.0064 \\ -0.006 \\ -0.0101 \\ 0.0162 \end{pmatrix}$$

至此,完成了在 RHadoop 环境下的线性回归分析。

12.2 Logistic 分析

Logistic 逻辑回归属于概率型非线性回归,它研究的是二分类观察结果与一些影响因素之间关系的一种多变量分析方法,本质上可与多重线性回归一样属于广义线性模型。在广义线性模型这一家族中的模型形式基本上都是差不多的,不同的就是因变量不同,如果是连续的,就是多重线性回归;如果是二项分布,就是 Logistic 回归;如果是 Poisson 分布,就是 Poisson 回归;如果是负二项分布,就是负二项回归,等等。只要注意区分它们的因变量就可以了。

Logistic 回归的因变量可以是二分类的,也可以是多分类的,但是二分类的更为常用,也更加容易解释。所以实际中最为常用的就是二分类的 Logistic 回归。

12.2.1 Logistic 分析原理

考虑具有 n 个独立变量的向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$,设条件概率 $P(\boldsymbol{y}=1|\boldsymbol{x})=p$ 为根据观测值相对于某事件 x 发生的概率。那么 Logistic 回归模型可以表示为

$$P(y=1\mid x)=\pi(x)=\frac{1}{1+\mathrm{e}^{-g(x)}}$$

这里 $f(x)=\dfrac{1}{1+\mathrm{e}^{-g(x)}}$称为 Logistic 函数。其中,$g(x)=w_0+w_1x_1+\cdots+w_nx_n$,那么在 x 条件下 y 不发生的概率为

$$P(y=0\mid x)=1-P(y=1\mid x)=1-\frac{1}{1+\mathrm{e}^{-g(x)}}=\frac{1}{1+\mathrm{e}^{g(x)}}$$

所以事件发生与不发生的概率之比为

$$\frac{P(y=1\mid x)}{P(y=0\mid x)}=\frac{p}{1-p}=\mathrm{e}^{g(x)}$$

这个比值称为事件的发生比,对发生比取对数得到

$$\ln\left(\frac{p}{1-p}\right)=g(x)=w_0+w_1x_1+\cdots+w_nx_n$$

可以看出 Logistic 回归都是围绕一个 Logistic 函数展开的,这个 Logistic 函数也被称为 Logit 转换,接下来就讲解一下如何用极大似然估计去求解回归模型中的参数。

假设有 m 个观测样本,观测值分别为 $y_1,y_2,\cdots,y_m$,设 $p_i=P(y_i=1|x_i)$为给定条件下得到 $y_i=1$ 的概率,同样地,$y_i=0$ 的概率为 $P(y_i=0|x_i)=1-p_i$,所以得到一个观测值的概率为 $P(y_i)=p_i^{y_i}(1-p_i)^{1-y_i}$。因为各个观测样本之间相互独立,所以它们的联合分布为各边缘分布的乘积。得到似然函数为

$$L(w)=\prod_{i=1}^{m}(\pi(x_i))^{y_i}(1-\pi(x_i))^{1-y_i}$$

然后,我们的目标是求出使这一似然函数的值最大的参数估计,最大似然估计就是求出参数 $w_0,w_1,\cdots,w_n$,使得 $L(w)$取得最大值。对函数 $L(w)$取对数得到

$$\ln L(w)=\sum_{i=1}^{m}(y_i\ln[\pi(x_i)]+(1-y_i)\ln[1-\pi(x_i)])$$

继续对这 $n+1$ 个 w_i 进行求偏导，得到 $n+1$ 个方程。比如现在对参数 w_k 求偏导，得到

$$\frac{\partial \ln L(w_k)}{\partial w_k} = \sum_{i=1}^{m} x_{ik}[y_i - \pi(x_i)] = 0$$

这样问题就转化为解这 $n+1$ 个方程形成的方程组。方程的求解一般不能人工计算，需要用到复杂的迭代方法，如牛顿—拉菲森迭代法。可以用软件编程求解，下面就来讲解一下如何在 RHadoop 的环境下编程求解 Logistic 回归，并附上 R 环境下的求解情况与之对比。

12.2.2 Logistic 分析案例

本例来自于 John Maindonald 所著的《Data Analysis and Graphics Using R》一书，其中所用的数据集是 anesthetic。数据集来自于一组医学数据，主要是调查了 30 例患者在做手术之前的 15 分钟给予预定水平的麻醉剂，观察患者在这 15 分钟的表现。是否有如肌肉抽搐和身体扭曲等的表现。如果有上述表现统称为有移动，其中变量 conc 表示麻醉剂的用量，move 则表示手术病人是否有所移动，而用 nomove 作为因变量，因此研究的重点在于 conc 的增加是否会使 nomove 的概率增加。下面将在 RHadoop 的环境下编程实现。

(1) 加载相应的包。

```
library(rJava)
library(rmr2)
```

(2) 构建数据框存储数据，并读入 hdfs 中。

```
anestot <- data.frame(conc = c(0.8,1.0,1.2,1.4,1.6,2.5),
                      move = c(6,4,2,2,0,0),
                      nomove = c(1,1,4,4,4,2),
                      total = c(7,5,6,6,4,2),
                      prop = c(0.1428,0.2000,0.6667,0.6667,0.9999,0.9999))
anestot <- to.dfs(anestot)
```

(3) 定义 map 函数，作为在 mapreduce 中的准备。

```
mapper1 <- function(.,data){
           y = log(data[,5]/(1 - data[,5]))
           x = data[,3]
           keyval(1,list(t(x) % * % y))
                  }
mapper2 <- function(.,data){
           x = data[,3]
           keyval(1,list(t(x) % * % x))
                  }
```

（4）运行 Mapreduce 并输出结果。

```
MP1 <- mapreduce(input = anestot, map = mapper1, combine = TRUE)
Y <- values(from.dfs(MP1))[[1]]
MP2 <- mapreduce(input = anestot, map = mapper2, combine = TRUE)
X <- values(from.dfs(MP2))[[1]]
```

（5）用 solve 函数求解出方程的系数。

```
solve(X,Y)
```

最终得到 Logistic 方程为

$$\ln\left(\frac{\text{prop}}{1-\text{prop}}\right)=1.067\text{conc}$$

至此，完成了在 RHadoop 中 Logistic 回归的实现。

12.3 判别分析

判别分析是一种统计分析方法，该方法是多变量统计分析中用于在已知一些对象分类的情形下确定新对象所属的类型。判别分析的理论依据是根据若干个指标的实际数据对已知分类的样本建立判别函数，从而预测新对象的分类。判别函数指的是一种统计模型，是对新样本和已知分类样本的相似程度的度量。要判别新样本的类型，除了判别函数外，还需要建立判别规则。根据判别规则的不同，可以将判别分析分成两类：Fisher 判别和 Bayes 判别。Fisher 判别分析的判别规则是确定性的，而 Bayes 判别分析的判别规则是统计性的，判别新对象类型时涉及概率性质。线性判别分析属于 Fisher 判别分析，是一种确定性判别方法，在对新对象进行归类时一般只需要考虑判别函数。

12.3.1 线性判别分析原理

线性判别分析(Linear Diseriminant Analysis，LDA)是一种经典的特征提取方法，该方法从高维特征空间中提取出最具代表性的低维特征，通过线性变换将研究对象映射到低维空间，使得不同类别的样本尽量分开，相同类别的样本尽可能聚集。

线性判别分析算法原理：

设 $X_{(j)}^{(i)}=(x_{j1}^{(i)},x_{j2}^{(i)},\cdots,x_{jk}^{(i)})^{\mathrm{T}},i=1,2,\cdots,c;\ j=1,2,\cdots,n_i$ 为已知分类的样本，其中 c 表示样本分类数，n_i 表示第 i 类样本的数目，k 表示每个对象的指标变量。

（1）第 i 类样本的均值：$\mu^{(i)}=\dfrac{1}{n_i}\sum\limits_{x\in dassi}x^{(i)}$

（2）总体样本的均值：$\mu=\dfrac{1}{N}\sum\limits_{i=1}^{c}\sum\limits_{j=1}^{n_i}x_{(j)}^{(i)}$

（3）类间离散度矩阵：$\boldsymbol{S}_b=\sum\limits_{i=1}^{c}n_i(\mu^{(i)}-\mu)(\mu^{(i)}-\mu)^{\mathrm{T}}$

(4) 类内离散度矩阵：$\boldsymbol{S}_w = \sum_{i=1}^{c}\sum_{j=1}^{n_i}(x_{(j)}^{(i)} - \mu^{(i)})(x_{(j)}^{(i)} - \mu^{(i)})^{\mathrm{T}}$

线性判别分析就是要选取使得样本类间离散度与类内离散度的比值达到最大的特征。Fisher 线性判别的准则函数定义如下：

$$J(\omega) = \arg\max_{\omega} \frac{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega}}{\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega}} = [\omega_1, \omega_2, \cdots, \omega_d]$$

通过最大化准则函数，即计算使得 $J(\omega)$ 取得极大值的向量 $\boldsymbol{\omega}^*$。$\boldsymbol{\omega}^*$ 的求解一般运用 Lagrange 乘子法，具体的求解过程如下：

令 $\boldsymbol{\omega}^{\mathrm{T}}S_w\boldsymbol{\omega} = a$，其中 a 为非零常数，Lagrange 函数定义如下：

$$L(\omega,\lambda) = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_w\boldsymbol{\omega} - \lambda(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{S}_b\boldsymbol{\omega} - a)$$

其中 λ 为 Lagrange 乘子。对上式的 ω 求偏导：

$$\frac{\partial L(\boldsymbol{\omega},\lambda)}{\partial \boldsymbol{\omega}} = \boldsymbol{S}_b\boldsymbol{\omega} - \lambda\boldsymbol{S}_w\boldsymbol{\omega}$$

令偏导数为 0，则 $\boldsymbol{S}_b\boldsymbol{\omega}^* - \lambda\boldsymbol{S}_w\boldsymbol{\omega}^* = 0$，即 $\boldsymbol{S}_b\boldsymbol{\omega}^* = \lambda\boldsymbol{S}_w\boldsymbol{\omega}^*$，其中 $\boldsymbol{\omega}^*$ 是 $J(\boldsymbol{\omega})$ 的极值解。

如果 $\boldsymbol{S}_w$ 非奇异，即可逆，上式两边左乘 $\boldsymbol{S}_w^{-1}$，得到 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{\omega}^* = \lambda\boldsymbol{\omega}^*$，则要计算 $J(\boldsymbol{\omega})$ 的极值 $\boldsymbol{\omega}^*$，就转换成了求解矩阵 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 的特征值问题，$\boldsymbol{\omega}^*$ 即为 $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ 最大特征值对应的特征向量。至此求解得到了线性判别函数的系数，即判别系数，得到判别函数，再建立临界值作为判别的标准，完成对样本的判别归类。值得注意的是，进行线性判别分析必须保证类内离散度矩阵 $\boldsymbol{S}_w$ 非奇异。

12.3.2 线性判别分析案例

线性判别分析法的一般步骤如下：

(1) 计算每类样本的均值、总体样本的均值。

(2) 计算类间离散度矩阵、类内离散度矩阵。

(3) 建立线性判别的准则函数，确定判别函数的系数，得到判别函数。

(4) 建立临界值，确定判别标准。

(5) 根据判别函数及判别标准对新样本进行判别归类。

该算法的 RHadoop 实现：

(1) 加载包，初始化 dfs。

```
library("rhdfs")
hdfs.init()
library("rmr2")
```

(2) 加载已知分类的数据，本例数据来自王斌会所著的《多元统计分析及 R 语言建模》一书，包括 20 个关于天气的对象，每个对象包含两个指标：湿温差和气温差，根据这两个指标将对象分成了两类：雨天(1)和晴天(2)。

```
X<-matrix(c(-1.9,-6.9,5.2,5.0,7.3,6.8,0.9,-12.5,1.5,3.8,0.2,-0.1,0.4,2.7,2.1,-
4.6,-1.7,-2.6,2.6,-2.8,3.2,0.4,2.0,2.5,0.0,12.7,-5.4,-2.5,1.3,6.8,6.2,7.5,14.6,
8.3,0.8,4.3,10.9,13.1,12.8,10.0),nrow=20,ncol=2)
```

```
c <- c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2)
Y <- cbind(X,c)
```

(3) 定义 mapper1 函数,对数据进行线性判别分析,得到线性判别函数系数。该函数包括总体样本均值、类样本均值、类间离散矩阵、类内离散矩阵及判别函数系数的计算方法的实现。

```
mapper1 = function(.,Y){
  Y1 <- Y[,-ncol(Y)]
  c <- Y[,ncol(Y)]
  uc <- unique(c)
  u <- apply(Y[,-ncol(Y)],2,mean) #总体样本均值
…
  keyval(1,V1)#线性判别函数系数 V1
}
```

(4) 定义 mapper2 函数,得到线性判别的判别效果。该函数根据得到的判别函数对原始数据进行重新分类,得到判别的正确率,从而确定线性判别的判别效果。

```
mapper2 = function(.,Y){
  Y1 <- Y[,-ncol(Y)]
  D <- Y1 %*% V1
  Newc <- c()
  for(b in 1:nrow(D)){
    if(D[b,]-mean(D)>=0) Newc[b] <- 1 else Newc[b] <- 2
  }#新的分类
…
keyval(1,qr)#判别的正确率
}
```

(5) 加载需要判别分类的数据。

```
NewX <- matrix(c(-1.8,0.3,-0.2,3.3,6.3,7.6),3,2)
```

(6) 定义 mapper3 函数,对数据进行判别分类。该函数实现了对未知类别的数据进行判别分类。

(7) 调用 mapper1 函数,得到线性判别系数。

```
mp1 <- mapreduce(input = to.dfs(Y),map = mapper1,combine = TRUE)
V1 <- values(from.dfs(mp1))
V1 #线性判别系
```

（8）调用 mapper2 函数，得到线性判别分析的判别效果。

```
mp2 <- mapreduce(input = to.dfs(Y),map = mapper2,combine = TRUE)
qr <- values(from.dfs(mp2))
qr #判别的正确率
```

（9）调用 mapper3 函数，得到线性判别分析结果。

```
mp3 <- mapreduce(input = to.dfs(NewX),map = mapper3,combine = TRUE)
NXc <- values(from.dfs(mp3))
NXc #判别分类结果
```

线性判别分析算法结果分析：

（1）已知分类的数据（数据来源于王斌会所著的《多元统计分析及 R 语言建模》一书）。

对象	湿温差	气温差	分类(1 雨天/2 晴天)
1	−1.9	3.2	1
2	−6.9	0.4	1
3	5.2	2.0	1
4	5.0	2.5	1
5	7.3	0.0	1
6	6.8	12.7	1
7	0.9	−5.4	1
8	−12.5	−2.5	1
9	1.5	1.3	1
10	3.8	6.8	1
11	0.2	6.2	2
12	−0.1	7.5	2
13	0.4	14.6	2
14	2.7	8.3	2
15	2.1	0.8	2
16	−4.6	4.3	2
17	−1.7	10.9	2
18	−2.6	13.1	2
19	2.6	12.8	2
20	−2.8	10.0	2

（2）线性判别系数：0.4183207，−0.9082994。

（3）线性判别分析的判别效果：判别的正确率为 90%。

（4）需要判别分类的数据。

对象	湿温差	气温差
1	−1.8	3.3
2	0.3	6.3
3	−0.2	7.6

（5）线性判别分析结果。

对象	分类
1	1
2	2
3	2

12.4 聚类分析

聚类指的是将数据集中的数据点分别划分到不同的类中，使得类内数据之间的相似度尽可能的大，不同类之间的相似性尽可能达到最小。K-均值（K-means）聚类算法是一种基于划分的聚类算法，该算法中的相似度是由一个类中的所有数据对象的平均值来计算得到的。该算法是 J. B. MacQueen 在 1967 年提出的，是一种解决聚类分析问题的经典算法，应用广泛，主要用于数据挖掘、机器学习及模式识别等领域。

12.4.1 K-means 聚类分析原理

K-均值聚类算法的基本思想如下：通过反复迭代，从而对数据集进行聚类，每次迭代完毕判断算法结束的条件，如果算法满足结束条件则迭代过程结束，产生聚类结果。

该算法通常采用聚类误差平方和函数作为聚类准则函数。假设样本集合 $A=\{p_1, p_2, \cdots, p_n\}$，它被聚类成 k 个类 $C_1, C_2, \cdots, C_k$，每个簇类 C_i 中分别包含 $\{n_1, n_2, \cdots, n_k\}$ 个数据对象，则误差平方和准则函数定义为：

$$J_c=\sum_{j=1}^{k}\sum_{i=1}^{n_j}\|x_i-c_j\|^2$$

其中，c_j 表示第 j 类 C_j 的样本均值，$c_j=\frac{1}{n_j}\sum_{i=1}^{n_j}t_i^j, i=1,2,\cdots,n_j$。

该函数刻画了类内数据之间的聚合程度，该值越小则说明聚类结果中各个类内的聚合程度越高。

另一种是类间距离和准则：

$$J=\sum_{j=1}^{k}(c_j-c)^T(c_j-c)$$

其中，c_j 表示聚类结果中类 C_j 的中心点，$j=1,2,\cdots,k$；c 表示数据集合 A 中所有数据对象的中心。

该函数刻画了各个类之间的分离程度，该值越大则说明各个类之间的分离性越好。

根据误差平方和准则及类间距离和准则来判断一个聚类结果聚类的效果，如果类内的聚合程度高，不同类之间的分离程度越高，则此聚类结果越好。

12.4.2 K-means 聚类分析案例

K-means 聚类算法的一般步骤如下：

(1) 指定初始聚类中心。在数据集中随机选定 k 个数据对象作为聚类中心，记为$(c_1, c_2, \cdots, c_k)$。

(2) 聚类。对于数据集中的每个样本 x_i，计算其与 k 个聚类中心的距离，根据距离最近原则将其归到距离它最近的中心代表的类中。

(3) 更新聚类中心。将步骤(2)中得到的 k 个类重新计算其中心：$c_j = \frac{1}{n_j}\sum_{i=1}^{n_j} t_i^j, i = 1, 2, \cdots, n_j$，即第 j 类 C_j 的样本均值。

(4) 误差平方和计算：$J_c = \sum_{j=1}^{k}\sum_{i=1}^{n_j} \| x_i - c_j \|^2$，即误差平方和准则函数。

(5) 判断。如果各个类的中心不再变化，或者满足规定的收敛准则，那么迭代结束，得到聚类结果；否则转到步骤(2)，继续迭代，直到满足迭代终止条件。

K-means 聚类算法具有算法思想简单，易于实现等优点，广泛应用于理论的研究及实际的应用中。

K-means 聚类算法的 Rhadoop 实现：

(1) 加载包，初始化 dfs。

```
library("rhdfs")
hdfs.init()
library("rmr2")
library(xlsx)
```

(2) 设定聚类的个数 k，迭代次数 num，读取数据，该样本数据来自 CSMAR 数据库中 2012 年全国 31 个省、市、自治区的居民收入与消费水平数据，同 6.3.3 节数据表。本例将 31 个省、市、自治区根据其居民收入和消费水平归为 4 类，进行 6 次迭代。

```
k = 4 # 聚类个数
num = 6 # 聚类迭代次数
juldata <- read.xlsx("/home/limin/data.xlsx",1,header = T)
X <- juldata[,c(4:10)]
A <- as.matrix(X)
```

(3) 定义 mapper1 函数，随机选定 k 个对象作为聚类中心，进行第一次聚类。

该函数实现了以任意 k 个对象为聚类中心的聚类分析。

```
mapper1 = function(.,A){
c <- sample(1:nrow(A),k,replace = F)
A1 <- A[c,]              #选取任意 k 个对象
…
  keyval(1,nearest)      #第一次聚类结果
}
```

(4) 定义 mapper2 函数,根据新的聚类中心进行聚类。该函数实现了 K-means 聚类分析。

(5) 调用 mapper1 函数,产生第一次聚类结果。

```
mp1 <- mapreduce(input = to.dfs(A),map = mapper1,combine = TRUE)
c2 <- values(from.dfs(mp1))
c2  #第一次聚类结果
```

(6) 调用 mapper2 函数,进行 num 次迭代,得到最终聚类结果。

```
for(a in 2:num){
mp2 <- mapreduce(input = to.dfs(A),map = mapper2,combine = TRUE)
  c2 <- values(from.dfs(mp2))
}
kc <- cbind(as.matrix(1:nrow(A),ncol = 1),as.matrix(values(from.dfs(mp2))))
kc  #最终聚类结果
```

聚类分析结果如表 12.1 所示。

表 12.1　K-means 聚类分析结果

地区	归类	地区	归类	地区	归类
北京	4	浙江	4	海南	1
天津	3	安徽	2	重庆	3
河北	2	福建	3	四川	2
山西	2	江西	1	贵州	1
内蒙古	3	山东	4	云南	1
辽宁	2	河南	2	西藏	1
吉林	1	湖北	2	陕西	2
黑龙江	1	湖南	2	甘肃	1
上海	4	广东	4	青海	1
江苏	4	广西	1	宁夏	1
				新疆	1

由表 12.1 可以看出,第一类:吉林、黑龙江、江西、广西、海南、贵州、云南、西藏、甘肃、青海、宁夏、新疆;第二类:河北、山西、辽宁、安徽、河南、湖北、湖南、四川、陕西;第三类:天津、内蒙古、福建、重庆;第四类:北京、上海、江苏、浙江、山东、广东。归类结果与地区实际收入和消费水平一致,验证了结果的正确性。

12.5 主成分分析

自 Hotlling 于 1933 年首先提出主成分分析以来，国内外都非常重视对该方法的研究与应用，并取得了相当大的进展，其应用成果已渗透到许多领域。

12.5.1 主成分分析原理

主成分分析法是利用降维的思想，在损失很少信息的前提下把多个指标根据一定原则和实际需要转化为几个综合指标的多元统计方法。该方法的基本思想是在保留尽可能多的原始信息的前提下达到降维的目的，从而简化问题的复杂性，抓住问题的主要矛盾。

主成分分析的推导过程：

设 $\boldsymbol{X}=(X_1,\cdots,X_p)'$ 为一个 p 维随机向量，其均值向量为 $\boldsymbol{\mu}=E(\boldsymbol{X})$，协方差为 $\Sigma=D(\boldsymbol{X})$。线性变换 $\boldsymbol{Y}=\boldsymbol{T}'\boldsymbol{X}$，其中 $\boldsymbol{Y}=(Y_1,Y_2,\cdots,Y_p)'$，$\boldsymbol{T}=(T_1,T_2,\cdots,T_p)$。确定一组新的变量 $Y_1,Y_2,\cdots,Y_m(m\leqslant p)$ 充分反应原始变量 $X_1,X_2,\cdots,X_p$ 的信息，且互不相关。

第一主成分：$Y_1=T'_1\boldsymbol{X}$，满足 $T'_1T_1=1$，使得 $D(Y_1)=T'_1\Sigma T_1$ 达到最大。

第二主成分：$Y_2=T'_2\boldsymbol{X}$，满足 $T'_2T_2=1$，使得 $D(\boldsymbol{Y}_2)=T'_2\Sigma T_2$ 达到最大。第一主成分是投影值的协方差达到最大的方向；第二主成分与第一主成分方向正交，使投影在该方向上的值的协方差达到最大。

第 k 主成分：$Y_k=T'_k\boldsymbol{X}$，满足 $T'_kT_k=1$，且 $\text{Cov}(Y_k,Y_i)=\text{Cov}(T'_k\boldsymbol{X},T'_i\boldsymbol{X})=0(i<k)$，使得 $D(Y_k)=T'_k\Sigma T_k$ 达到最大。

其中，第一主成分 $Y_1=T'_1\boldsymbol{X}$ 综合原始变量 $X_1,X_2,\cdots,X_p$ 的能力最强，而 $Y_2,\cdots,Y_p$ 的综合能力依次递减。

综上所述，若 $\boldsymbol{X}$ 的协方差 Σ 的特征值为 $\lambda_1\geqslant\lambda_2\geqslant\cdots\geqslant\lambda_p\geqslant0$，相应的单位化特征向量为 $T_1,T_2,\cdots,T_p$。则由此所确定的主成分为 $Y_1=T'_1\boldsymbol{X},Y_2=T'_2\boldsymbol{X},\cdots,Y_m=T'_m\boldsymbol{X}$，其方差分别为 Σ 的特征根。为了保持信息不丢失，$\boldsymbol{Y}$ 的各分量方差和与 $\boldsymbol{X}$ 的各分量方差和相等。

第 k 个主成分 Y_k 的贡献率：

$$\varphi_k=\frac{\lambda_k}{\sum_{k=1}^{p}\lambda_k}$$

即第 k 个方差在全部 p 个方差中的比重称为第 k 个主成分的贡献率。贡献率越大，新变量 Y_k 综合原始变量 $X_1,X_2,\cdots,X_p$ 的能力越强。

若只取 $m(<p)$ 个主成分，则主成分 $Y_1,Y_2,\cdots,Y_m$ 的累积贡献率：

$$\phi_m=\frac{\sum_{k=1}^{m}\lambda_k}{\sum_{k=1}^{p}\lambda_k}$$

即为 $Y_1,Y_2,\cdots,Y_m$ 综合 $X_1,X_2,\cdots,X_p$ 的能力。进行主成分分析减少指标数，确定主成分是一个很实际的问题。选取主成分通常以累积贡献率达到 85%以上的标准确定 m 的取值，即

$$\frac{\sum_{k=1}^{m}\lambda_k}{\sum_{k=1}^{p}\lambda_k} \geqslant 85\%$$

这样既尽量减少了信息的损失，也达到了降维的目的。大多数情况下，当 $m=3$ 时就可使选取的主成分累积贡献率达到 85%以上。

通常选择评价指标体系后，运用对各指标加权的方法进行综合，得到综合评价得分，从而确定综合排名。利用主成分进行综合评价时，要充分利用原始变量的信息，将原始指标进行综合。由于方差贡献率反映了各主成分的信息量，因此对主成分进行加权综合，通常根据方差贡献率确定主成分的权重系数。

设 $Y_1, Y_2, \cdots, Y_m$ 是所求的 m 个主成分，其特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_m$，则有

$$w_i = \frac{\lambda_i}{\sum_{i=1}^{m}\lambda_i}, \quad i = 1, 2, \cdots, m$$

则综合评价函数：

$$Z = w_1 Y_1 + w_2 Y_2 + \cdots + w_m Y_m.$$

12.5.2　主成分分析案例

主成分分析法是一种客观赋权的方法，该方法充分利用原始变量的全部信息，根据方差贡献率确定主成分的权重系数，全面客观地反映指标的重要程度，使综合评价结果更合理。

主成分分析综合评价的一般步骤如下：

(1) 建立指标体系。

(2) 进行指标间相关性判定及数据处理，确立相关系数矩阵或协方差矩阵。

(3) 计算相关系数矩阵或协方差矩阵的特征值与特征向量。

(4) 选取主成分，确定指标权重。

(5) 确定主成分表达式。

(6) 计算综合评价值，做出综合评价。

主成分分析算法的 Rhadoop 实现：

(1) 加载包，初始化 dfs。

```
library("rhdfs")
hdfs.init()
library("rmr2")
library("xlsx")
```

(2) 数据读取，PCAFAdata.xlsx 存储于 Hadoop 中/home/limin/文件夹下。该样本数据来自 CSMAR 数据库中 2012 年全国 31 个省、市、自治区的居民收入与消费水平数据，同 6.3.4 节数据表。

```
pcadata <- read.xlsx("/home/limin/PCAFAdata.xlsx",1,header = T)
X <- pcadata[,c(4:10)]
```

（3）定义 mapper1 函数，对原始数据进行标准化处理。

该函数实现了数据的标准化处理，消除了量纲的影响。

```
mapper1 = function(.,X){
  mju <- c()
  sigma <- c()
  for(j in 1:ncol(X)){
    mju[j] <- mean(X[,j])          #均值
    sigma[j] = sqrt(sd(X[,j]))     #标准差
  }
 …
  keyval(1,Y)                      #标准化矩阵
}
```

（4）定义 mapper2 函数，进行主成分分析综合评价。该函数包括成分贡献率、成分载荷矩阵、主成分得分矩阵、综合得分及排名的算法实现。

```
mapper2 = function(.,Y){
  R <- cov(Y)
  ev <- eigen(R)
  E1 <- ev$val[order(-ev$val)]     #特征值由大到小排列
  H <- order(-ev$val)
  s <- 0
  n1 = length(E1)
  for(i in 1:n1){
   s <- s + E1[i]
  }
  g <- E1/s #成分贡献率
  …
  keyval(1,rbind(p1,Or)) #综合排名
  }
```

（5）调用 mapper1 函数，得到原始数据标准化后的矩阵。

```
mp1 <- mapreduce(input = to.dfs(X),map = mapper1,combine = TRUE)
Y <- values(from.dfs(mp1))
Y     #标准化后的矩阵
```

（6）调用 mapper2 函数，得到主成分个数及主成分分析综合评价的排名。

```
mp2 <- mapreduce(input = to.dfs(Y),map = mapper2,combine = TRUE)
p <- values(from.dfs(mp2))[1,1]
p  #主成分个数
Order <- cbind(as.matrix(1:nrow(X),ncol = 1),as.matrix(values(from.dfs(mp2))[2,]))
Order  #综合排名
```

(1) 主成分个数：2。

(2) 主成分分析综合排名如表 12.2 所示。

表 12.2 主成分分析综合排名

地区	r	地区	r	地区	r
北京	3	浙江	4	海南	27
天津	7	安徽	17	重庆	16
河北	11	福建	9	四川	10
山西	19	江西	22	贵州	28
内蒙古	15	山东	6	云南	24
辽宁	8	河南	12	西藏	31
吉林	21	湖北	13	陕西	18
黑龙江	23	湖南	14	甘肃	29
上海	1	广东	2	青海	30
江苏	5	广西	20	宁夏	26
				新疆	25

可以看到结果与 6.3.4 节中用 R 中内置函数进行主成分分析的结果是一致的，也进一步验证了结果的正确性。

12.6 因子分析

因子分析(Factor Analysis)是一种针对多个变量的统计分析方法。该方法是 Charler Spearman 于 1904 年提出来的，如今广泛应用于统计、经济、物流及生物医学等领域。因子分析探讨的是多个变量之间的相互依赖关系，并把多个变量转换为少数几个公共因子，是主成分分析方法的推广。

12.6.1 因子分析原理

因子分析模型(正交因子模型)：

设 $\boldsymbol{X}=(X_1,X_2,\cdots,X_p)'$ 是可观测的随机向量，$E(\boldsymbol{X})=\mu$，$D(\boldsymbol{X})=\Sigma$。且设 $\boldsymbol{F}=(F_1,F_2,\cdots,F_m)'(m<p)$ 是不可观测的随机向量，$E(\boldsymbol{F})=0$，$D(\boldsymbol{F})=I_m$，即分量方差为 1，且互不相关。又设 $\boldsymbol{\varepsilon}=(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_p)'$ 与 $\boldsymbol{F}$ 互不相关，且

$$E(\boldsymbol{\varepsilon})=0,\quad D(\boldsymbol{\varepsilon})=diag(\sigma_1^2,\sigma_2^2,\cdots,\sigma_p^2)\hat{=}D$$

假定随机向量 X 满足以下模型：

$$\begin{cases} X_1-\mu_1=a_{11}F_1+a_{12}F_2+\cdots+a_{1m}F_m+\varepsilon_1 \\ X_2-\mu_2=a_{21}F_1+a_{22}F_2+\cdots+a_{2m}F_m+\varepsilon_2 \\ \vdots \\ X_p-\mu_p=a_{p1}F_1+a_{p2}F_2+\cdots+a_{pm}F_m+\varepsilon_p \end{cases}$$

此模型为正交因子模型，用矩阵表示为

$$\boldsymbol{X}=\mu+\boldsymbol{AF}+\boldsymbol{\varepsilon}$$

其中 $\boldsymbol{F}=(F_1,F_2,\cdots,F_m)'(m<p)$ 为 $\boldsymbol{X}$ 的公共因子；$\boldsymbol{\varepsilon}=(\varepsilon_1,\varepsilon_2,\cdots,\varepsilon_p)'$ 为 $\boldsymbol{X}$ 的特殊因子；公共因子对 $\boldsymbol{X}$ 的每一个分量 X_i 都有作用，而 ε_i 只对 X_i 起作用，而且各特殊因子之间及特殊因子与所有公共因子之间是互不相关的。

上述模型中的矩阵 $\boldsymbol{A}=(a_{ij})_{p\cdot m}$ 是待估的系数矩阵，为因子载荷矩阵。$a_{ij}(i=1,\cdots,p;j=1,\cdots,m)$ 为第 i 个变量在第 j 个因子上的载荷，即因子载荷。

在因子分析中，因子的个数 m 的选取通常采用确定主成分个数的原则，即满足

$$\frac{\sum_{k=1}^{m}\lambda_k}{\sum_{k=1}^{p}\lambda_k}\geqslant 85\%$$

的 m 值。

如果选取的 m 个因子不能很好地反映其实际意义，那么需要对因子进行旋转，使其具有比较明确的实际意义。根据原指标的线性组合对因子得分进行求解，常用的方法有回归估计法、Bartlett 估计法等。一般由公共因子的方差贡献率作为因子的权重系数，以得到综合得分。计算方法如下：

设 $F_1,F_2,\cdots,F_m$ 是所求的 m 个因子，其特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_m$，则有

$$w_i=\frac{\lambda_i}{\sum_{i=1}^{m}\lambda_i},\quad i=1,2,\cdots,m$$

则综合因子得分函数为

$$Z=w_1F_1+w_2F_2+\cdots+w_mF_m$$

12.6.2 因子分析案例

运用因子分析进行综合评价的一般步骤如下：

(1) 对原始数据进行标准化处理，消除变量之间数量级及量纲的影响。

(2) 对标准化数据进行处理，确立其相关矩阵或协方差矩阵。

(3) 计算相关矩阵或协方差矩阵的特征值与特征向量。

(4) 根据一定规则选取因子。

(5) 若选取的因子表示实际意义的效果不明显，则进行因子旋转。

(6) 根据原始指标的线性组合进行因子得分的求解。

(7) 确定各因子的权重。

(8) 计算综合因子评价值，做出综合评价。

因子分析算法的 Rhadoop 实现：

(1) 加载包，初始化 dfs。

```
library("rhdfs")
hdfs.init()
library("rmr2")
```

(2) 数据读取,样本数据来自 CSMAR 数据库,同 13.6.1 节数据。

```
fadata <- read.xlsx("/home/limin/PCAFAdata.xlsx",1,header = T)
Y <- fadata[,c(4:10)]
```

(3) 定义 mapper1 函数,对原始数据进行均值化处理。

该函数实现了对原始数据的均值化处理,消除了量纲的影响。

(4) 定义 mapper2 函数,进行因子分析综合评价。该函数包括了样本协方差阵、相关阵、方差贡献率、因子得分及综合得分排名的算法实现。

```
mapper2 = function(.,X){
  mju <- c()
  for(j in 1:ncol(X)){
    mju[j] <- mean(X[,j])
  }
Y <- t(t(X) - mju)
  s <- 0
  for(i in 1:nrow(X)){
    s <- s + Y[i,] % * % t(Y[i,])
  }
  cR <- s/(nrow(X) - 1)      #样本的协方差阵
…
  keyval(1,SOr)              #综合得分排名
}
```

(5) 调用 mapper1 函数,得到原始数据标准化后的矩阵。

```
mp1 <- mapreduce(input = to.dfs(Y),map = mapper1,combine = TRUE)
X <- values(from.dfs(mp1))
X      #标准化后的矩阵
```

(6) 调用 mapper2 函数,得到因子分析综合评价的排名。

```
mp2 <- mapreduce(input = to.dfs(X),map = mapper2,combine = TRUE)
rank <- values(from.dfs(mp2))
rank   #综合排名
```

因子分析算法综合评价分析结果如表 12.3 所示。

表 12.3 因子分析综合排名

地区	r	地区	r	地区	r
北京	2	浙江	3	海南	25
天津	5	安徽	18	重庆	11
河北	15	福建	7	四川	16
山西	24	江西	22	贵州	29

续表

地区	r	地区	r	地区	r
内蒙古	10	山东	8	云南	26
辽宁	9	河南	17	西藏	31
吉林	14	湖北	12	陕西	19
黑龙江	21	湖南	13	甘肃	30
上海	1	广东	4	青海	28
江苏	6	广西	20	宁夏	23
				新疆	27

可以看到结果与6.3.5节中用R中内置函数进行因子分析的结果是一致的，也进一步验证了结果的正确性。

12.7 商品推荐算法

随着电子商务规模的不断扩大，商品个数和种类快速增长，顾客需要花费大量的时间才能找到自己想买的商品。这种浏览大量无关的信息和产品的过程无疑会使淹没在信息过载问题中的消费者不断流失。为了解决这些问题，个性化推荐系统应运而生。个性化推荐是根据用户的兴趣特点和购买行为向用户推荐用户感兴趣的信息和商品，是建立在海量数据挖掘基础上的一种高级商务智能平台，以帮助电子商务网站为其顾客购物提供完全个性化的决策支持和信息服务。

在电子商务时代，商家通过购物网站提供了大量的商品，客户无法一眼通过屏幕就了解所有的商品，也无法直接检查商品的质量。所以，客户需要一种电子购物助手，能根据客户自己的兴趣爱好推荐客户可能感兴趣或者满意的商品。个性化推荐系统具有良好的发展和应用前景。目前，几乎所有的大型电子商务系统，如Amazon、eBay等不同程度地使用了各种形式的推荐系统。国内方面，知名购物网站麦包包、凡客诚品、库巴网、红孩子等都率先选择了本土最先进的百分点推荐引擎系统构建个性化推荐服务系统。在日趋激烈的竞争环境下，个性化推荐系统能有效地保留客户，提高电子商务系统的服务能力。因此，成功的推荐系统会带来巨大的效益。

12.7.1 商品推荐算法原理

推荐系统的算法有很多，如基于内容推荐、协同过滤推荐、基于关联规则推荐、基于知识推荐、组合推荐等。协同过滤推荐技术是推荐系统中应用最早和最为成功的技术。本文主要介绍基于协同过滤的推荐算法，它一般采用最近邻技术，利用用户的历史喜好信息计算用户之间的距离，进而利用目标用户的最近邻居用户对商品评价的加权评价值来预测目标用户对特定商品的喜好程度，然后系统根据这一喜好程度来对目标用户进行推荐。协同过滤算法的最大优点是对推荐对象没有特殊的要求，能处理非结构化的复杂对象，如音乐、电影等。

协同过滤首先基于这样一个假设：跟你喜好相似的人喜欢的东西你也很有可能喜欢。因此，为一个用户找到他真正感兴趣的内容的好方法是首先找到与此用户有相似兴趣的其

他用户,然后将他们感兴趣的内容推荐给此用户。其基本思想非常易于理解,在日常生活中,我们往往会利用好朋友的推荐来进行一些选择。协同过滤正是把这一思想运用到电子商务推荐系统中来,基于其他用户对某一内容的评价来向目标用户进行推荐。

12.7.2 商品推荐案例

基于用户的协同过滤算法主要分为三个步骤:

(1) 收集用户偏好。

(2) 找到相似的用户或物品。

(3) 计算推荐。

下面将在 RHadoop 环境下完成对商品推荐的编程。

本例所用的数据来自于 Sean Owen 等所著的《Mahout in Action》一书,数据集是来自 5 个顾客的调查数据(具体数据信息如下),其中变量 user 表示顾客信息,item 表示购买商品的信息,pref 表示购买商品的偏好信息。本例研究的重点是根据消费者购买商品的偏好信息,使用协同过滤算法产生推荐商品,下面将在 RHadoop 的环境下编程实现。

顾客购买商品及对商品的评价信息如下:

```
   user  item  pref
1     1   101   5.0
2     1   102   3.0
3     1   103   2.5
4     2   101   2.0
5     2   102   2.5
6     2   103   5.0
7     2   104   2.0
8     3   101   2.0
9     3   104   4.0
10    3   105   4.5
11    3   107   5.0
12    4   101   5.0
13    4   103   3.0
14    4   104   4.5
15    4   106   4.0
16    5   101   4.0
17    5   102   3.0
18    5   103   2.0
19    5   104   4.0
20    5   105   3.5
21    5   106   4.0
```

为了产生推荐,应该按照如下步骤进行:

(1) 计算相似矩阵。

(2) 建立用户得分矩阵。

(3) 产生推荐商品。

具体编程实现如下:

（1）安装并加载相应的包，本例用到的安装包是 Matrix。

```
library(rmr2)
library(Matrix)
```

（2）以数据框的形式载入数据，并把数据导入到 HDFS。

```
train <- data.frame(
user <- c(1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,5),
item <- c(101,102,103,101,102,103,104,101,104,105,107,101,
103,104,106,101,102,103,104,105,106),
pref <- c(5,3,2.5,2,2.5,5,2,2,4,4.5,5,5,3,4.5,4,4,3,2,4,3.5,4))
train.hdfs <- to.dfs(train)
```

（3）定义 mapper1 函数，计算相似矩阵。该函数实现了商品间相似矩阵的计算。

```
mapper1 = function(.,data){
    users = sort(unique(data $ user))
    items = sort(unique(data $ item))
    prefs = sparseMatrix(i = match(data $ user, users), j = match(data $ item, items), x = data
 $ pref)
…
keyval(1,co) #相似矩阵
                    }
```

（4）定义 mapper2 函数，获得商品推荐结果。该函数即为商品推荐函数，涉及 4 个参数：相似矩阵、先前购买什么商品、对该商品的得分是多少、推荐商品的数目，实现了商品推荐算法。

```
    mapper2 = function(.,data){
co = data[[1]]                          #商品相似矩阵
        k = data $ k                    #用户购买过的商品
        score = data $ score            #用户商品的评分
        m = data $ m                    #推荐商品的数目
 …
        keyval(1,recommend)             #推荐结果
                     }
```

（5）调用 mapper1 函数和 mapper2 函数，获取推荐商品的列表。

```
MP.co <- mapreduce(input = train.hdfs, map = mapper1, combine = TRUE)
co <- values(from.dfs(MP.co))
data <- to.dfs(list(co,k = c(1,3),score = c(1,5),m = 2))
MP.recommend <- mapreduce(input = data, map = mapper2, combine = TRUE)
recommend <- values(from.dfs(MP.recommend))
```

(6) 根据程序得出商品的相似矩阵及最后商品推荐得到的结果如下。

其中，相似矩阵为：

```
> round(co,3)
      1     2     3     4     5     6     7
1 0.000 0.028 0.035 0.033 0.016 0.029 0.012
2 0.028 0.000 0.057 0.021 0.027 0.030 0.020
3 0.035 0.057 0.000 0.026 0.016 0.027 0.014
4 0.033 0.021 0.026 0.000 0.039 0.047 0.024
5 0.016 0.027 0.016 0.039 0.000 0.027 0.074
6 0.029 0.030 0.027 0.047 0.027 0.000 0.017
7 0.012 0.020 0.014 0.024 0.074 0.017 0.000
```

得到的推荐结果为：

```
>recommend = values(from.dfs(MP.recommend))
>recommend
[[1]]
[1]"2""7"
[[2]]
       2        7
3.672986  3.129787
```

所以，最终在顾客购买商品的已有偏好商品1和商品3的情况下，对该顾客推荐商品2和商品7。

12.8　差异分析

随着经济的高速增长，我国出现了严重的贫富差距问题。目前，我国已经成为世界上贫富差距最大的国家。我国的贫富差距是指我国社会中个人财富不均衡的现象，即人们对物质生活资料占有的差距。贫富差距直观地表现为人们收入的差距，工资水平在一定程度上可以反映收入问题，判定一个地区工资水平的高低，不仅能反映出当地经济的发展水平和人民的生活水平，还能为国家制定宏观经济政策提供一定的参考。

12.8.1　多维标度法的原理

多维标度法(MDS)是在低维空间展示“距离”数据结构的多元数据技术。它可以把多维的数据降维到二维空间，从而实现数据的可视化。

古典多维标度法求解的一般步骤如下：

(1) 计算样品间的距离矩阵。

(2) 根据距离矩阵构造内积矩阵。

(3) 计算内积矩阵的特征值，并选取 r 个最大的特征值 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_r \geqslant 0$ 及其对应的单位特征向量。其中，r 的确定方法有两种：一是事先确定 $r=1,2$ 或 3；二是通过计算前 r 个大于零的特征值占全体特征值的比例 κ 确定。

$$\kappa = \frac{\lambda_1 + \lambda_2 + \cdots \lambda_r}{|\lambda_1| + |\lambda_2| + \cdots |\lambda_n|} \geqslant \kappa_0$$

其中，κ_0 为预先给定的变差贡献比例。

12.8.2 差异分析案例

在 RHadoop 环境下实现差异分析，样本数据来自国泰安 CSMAR 数据库，选取人口就业与工资数据库中 2013 年全国 31 个省、市、自治区的职工平均工资及指数，包括职工平均货币工资合计、国有单位职工平均货币工资、城镇集体单位职工平均货币工资、其他单位职工平均货币工资数据。

在 RHadoop 环境下的计算过程及结果分析如下：

（1）加载包。

```
library("rhdfs")
hdfs.init()
library("rmr2")
library("xlsx")
library("MASS")
```

（2）载入数据文件。

```
wage <- read.xlsx("/home/limin/chayi.xlsx",1,header = T)
wage <- wage[,c(4:7)]
data <- as.matrix(wage)
```

（3）定义 map 函数，求出各个城市之间的距离。该函数实现了各个城市间距离矩阵的计算。

```
map = function(.,data){
  D = dist(data, method = "euclidean",diag = T, upper = FALSE, p = 2)
…
keyval(1,X)     #距离矩阵
}
```

（4）调用 map 函数，得到城市间距离矩阵。

```
Mds <- mapreduce(input = to.dfs(data), map = map,combine = TRUE)
M <- from.dfs(Mds) $ val
```

（5）采用多维标度法把多维指标降维到二维空间，并作出图形。

```
fit <- isoMDS(M,k = 2)
x <- fit $ points[,1]
y <- fit $ points[,2]
par(mar = c(4,4,1,2),cex = 0.75)
```

```
plot(x,y);abline(v = 0,h = 0,lty = 3)
area = c('北京','天津','河北','山西','内蒙古','辽宁','吉林','黑龙江','上海','江苏','浙江','安徽
','福建','江西','山东','河南','湖北','湖南','广东','广西','海南','重庆','四川','贵州','云南','西
藏','陕西','甘肃','青海','宁夏','新疆')
text(x,y,labels = area,adj = - 0.01)
```

最后，输入图形如图 12.1 所示。

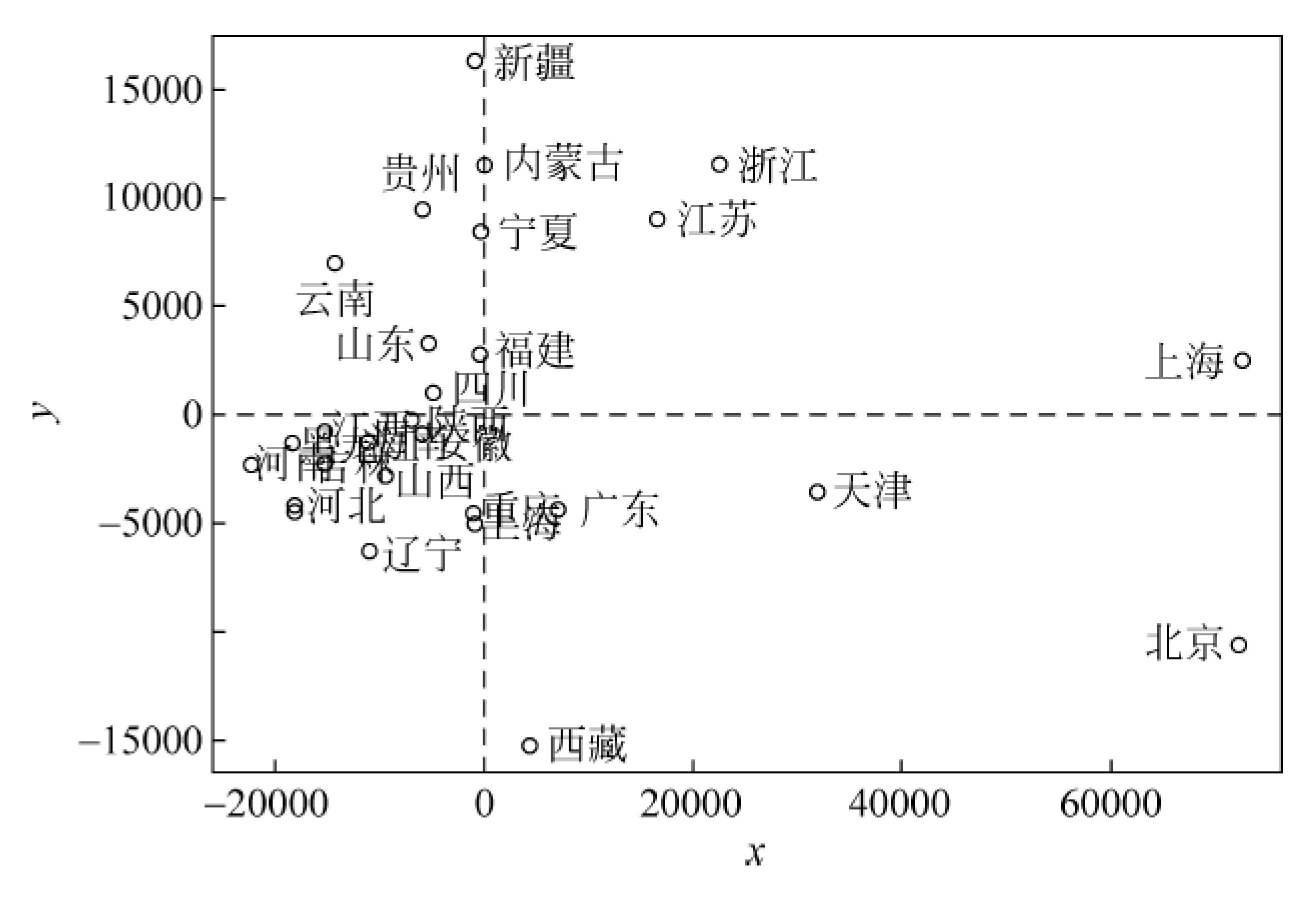


图 12.1　差异分析图

从图 12.1 可以看出，北京、上海的工资水平较相近，天津、浙江、江苏、广东的工资水平较相近，新疆、内蒙古、贵州、宁夏的工资水平较相近。因此，把每个地区的多维指标降维到二维空间更有利于比较它们的相对位置关系，这是数据可视化的一种方式。

APPENDIXES

附录一

国泰安CSMAR数据下载

本书绝大部分样本数据来自国泰安 CSMAR 数据库，下面给出了本书 6.3 节多元统计分析中聚类分析样本数据的获取下载步骤。

单击 http://www.gtarsc.com/进入国泰安数据服务中心登录页面，输入账号和密码，进入数据中心首页，如图 A.1 所示。



图 A.1 国泰安数据服务中心登录页面

CSMAR 数据库设置了两种数据查询方式：单表查询和自定义查询。用户可以根据自身需求选择不同的查询方式，如图 A.2 所示。

选择“自定义查询”，单击“区域经济”→“居民收入与消费”，依次选择农村居民消费水平、城镇居民消费水平、城镇居民家庭平均每人全年总收入、农村居民家庭平均每人全年纯收入、城镇居民消费支出、农村居民消费支出，城乡储蓄，时间选择为 2012 年，如图 A.3 所示。

单击下载数据，得到 2012 年全国 31 个省、市、自治区的居民收入与消费状况表。至此，完成了 CSMAR 数据的下载过程。

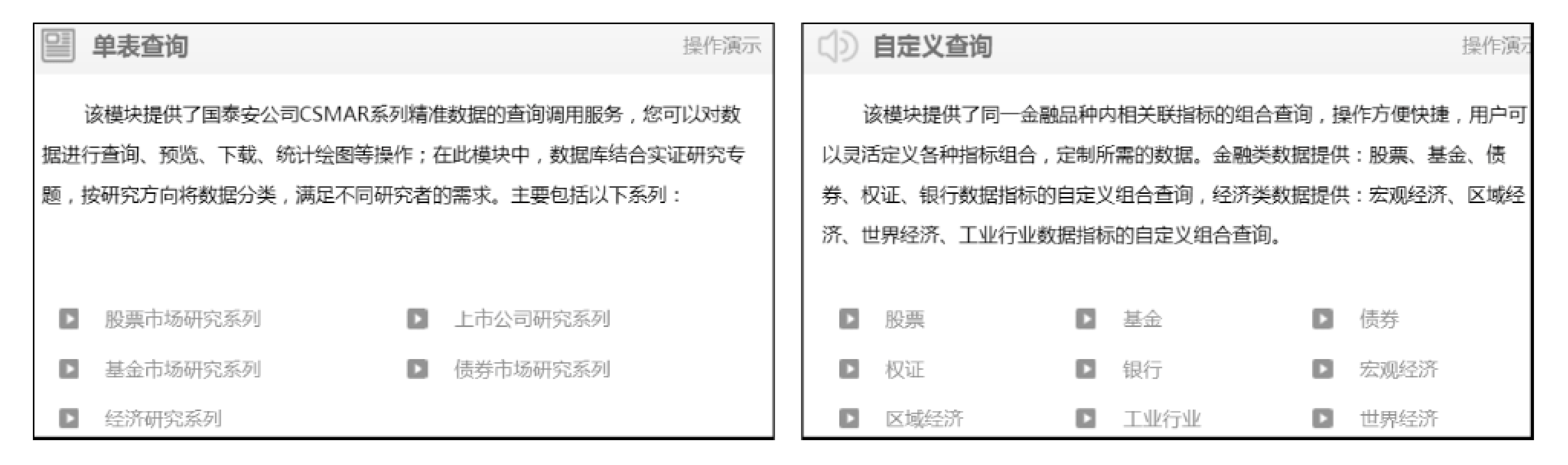


图 A.2 CSMAR 数据库查询方式



图 A.3 CSMAR 数据选择

APPENDIXES

附录二 深圳国泰安教育技术股份有限公司简介

深圳国泰安教育技术股份有限公司(以下简称为“国泰安”或“公司”)是一家为教育与投资业提供综合解决方案的国家级高新技术企业。自 2000 年以来,国泰安一直致力于为国内外教育和投资机构提供集“研究数据、专业实验、云平台建设、软硬件系统和增值服务”为一体的综合性解决方案。公司的产品与服务主要包括为高等教育、职业教育、基础教育领域提供教研、教学、管理、资源、实验及增值服务全方位支持的“易”系列教育服务,涵盖中国证券、期货、外汇、宏观、行业等领域的“元”系列精准数据服务,以及为金融机构提供全套量化投资服务方案的“宽”系列金融服务,对推动我国教育创新及金融创新做出了较大的贡献。

国泰安公司总部位于科技之都深圳,背靠国际经济、金融、贸易中心香港,并在北京、上海、广州、重庆、香港等 100 余个城市设有分公司或办事处,形成通达全国的服务网络。同时,国泰安业务已拓展到韩国、日本、新加坡、美国、澳洲、中国香港和中国台湾等 20 多个国家和地区,为全球 2000 多家教育机构、研究机构、金融机构客户提供创新服务。

国泰安拥有系统、专业的事业部体系,为包括高校、高职、中职、基础教育领域的 60 余个专业学科提供教学综合解决方案,涵盖金融财会、商贸管理、创业就业、物流会展、信息技术、工程制造、基础建设等专业,并得到学校的高度认可和广泛应用。同时,公司还在不断整合国内外优质教育资源,进一步丰富产品线,满足学校更多专业需求,为用户提供更多优秀的教学综合解决方案。

国泰安公司现拥有 3500 多位优秀员工,研发人员及技术工程师占公司人数的 60%。毕业于美国宾夕法尼亚大学、得克萨斯州立大学、香港理工大学、北京大学、清华大学等海内外名校的博士、硕士及海外留学人员超过公司人数的 30%。100 多位来自普林斯顿大学、香港大学、北京大学、清华大学、上海交通大学等国内外著名大学教授、权威学者组成国泰安顾问团队。30 余家海内外学术、业界翘楚(包括美国沃顿商学院、香港大学、日本 QUICK、韩国 EBSCO)与国泰安达成长期合作伙伴关系。

近 5 年来,国泰安研发投入 3 亿元以上,经营业绩实现连续复合增长超过 100%。至今已建立起规模庞大、分工精细的三大管理体系,40 多个事业部、60 余个专业学科的丰富产品

线，拥有近200项自主创新产品专利和著作权产品，完成了10余家企业并购整合，实现了企业跨越式发展。

2012年广东省电子信息（软件）自主创新产品认证的71个项目，国泰安独占3席；国泰安金融实验室与华为、大族激光等知名企业喜获深圳市重点自主创新产品殊荣；公司已通过CMMI三级认证、ISO9001：2008质量认证，且荣获了2013年中国年度创新软件产品（易教育平台V1.0）、广东省守合同重信用企业、计算机信息系统集成企业3级、国家火炬计划重点高新技术企业、国家规划布局内重点软件企业等重要资质。

国泰安在国内率先引进欧美国家先进的教学理念，并结合中国实际探索出一套系统、先进的教学综合解决方案，将传统的实验室建设提升为融“实验室建设、校企共建、资源共享、品牌提升”为一体的综合解决方案，确保每一次项目的完成均是一份完美的答卷。

目前，公司已经为美国、英国、法国、澳洲、日本、新加坡等20多个国家和地区的2000余家客户提供了卓越的产品与服务。在中国，为北京大学、清华大学、上海交通大学、厦门大学等知名高校和北京电子科技职业学院、深圳职业技术学院、厦门城市职业学院等职校提供综合实验解决方案，积累了丰富的建设经验，可为用户提供从实验室设计运营、教学资源建设、师资队伍培养、专业合作共建等全方位的服务和支持。

国泰安在行业内独家创新推出极受客户认可的丰富全面、周密精深、专属订制、面向未来的增值服务。依托国泰安四大专业服务中心：实验软件设备设计制造中心、职业教育实验服务中心、学生创业就业服务中心、校企合作服务中心，向客户提供包括专业共建、合作办学，科研、课题、论文学术合作，校际专业交流、资源共享，合作举办大型学术论坛、行业峰会，建立产学研校企合作联盟，学校品牌战略建设等增值服务，与客户一同打造品牌化、特色化的区域、国家乃至国际教育标杆。

读者可访问公司主页 http://www.gtaedu.com/13-3.html 获取更多详情。

参考文献

[1] 陶雪娇,胡晓峰,刘洋.大数据研究综述[J].系统仿真学报，2013,8(25)：143.
[2] 王妍,柴剑平.大数据及相关技术解读[J].广播电视信息，2014(2)：18-21.
[3] http://www.open-open.com/bbs/view/1404965109607.
[4] http://www.cfern.org/wjgg/wjggDisplay.asp? Id=2353.
[5] http://www.csdn.net/article/2014-05-26/2819939.
[6] 深圳国泰安教育技术股份有限公司大数据事业部群,中科院深圳先进技术研究院——国泰安金融大数据研究中心.大数据导论关键技术与行业应用最佳实践[M].北京：清华大学出版社,2015.
[7] 李明,王威扬,等译. R 与 Hadoop 大数据分析实战[M].北京：机械工业出版社,2014.
[8] 黄宜华.深入理解大数据、大数据处理与编程实践[M].北京：机械工业出版社,2014.
[9] http://baike.baidu.com/link? url = lnD8lmwKE4vGOneQhSBhNfFPNt7MfXl-sSyubVzcdYMN2Xsf9 ylWBOLSLZt0YpVWInArgZunuSpSgv6G2bGrI_.
[10] [美]Robert I. Kabacoff. R 语言实战[M].高涛,肖楠,陈钢,译.北京：人民邮电出版社,2013.
[11] http://blog.sina.com.cn/s/blog_744c2fb701014su8.html.
[12] http://www.biostatistic.net/thread-3228-1-1.html.
[13] http://zhidao.baidu.com/link? url = 42irj4SVrpp2mof7fz8AaM8HPukTDQsNO55YcV6LjklatYQlGsuOOMbo5ghZ-myTQRwiZvycHTi9fty-xJhRmK.
[14] 韩伟,毛俊杰. R 语言与商业智能[M].北京：电子工业出版社，2014.
[15] 李诗羽,张飞,王正林.数据分析：R 语言实战[M].北京：电子工业出版社,2014.
[16] 何晓群.多元统计分析.第 2 版[M].北京：中国人民大学出版社,2008.
[17] 王斌会.多元统计分析及 R 语言建模.第 2 版[M].广州：暨南大学出版社,2011.
[18] 汤勍.基于小波分析的金融时间序列研究与应用[D].中国科学技术大学,2011.
[19] Yanchang Zhao. R 语言与数据挖掘最佳实践和经典案例[M].陈健,黄琰,译.北京：机械工业出版社,2014.
[20] 张世英,柯珂. ARCH 模型体系[J].系统工程学报,2002,3(17)：236-244.
[21] 邹建军,张宗益,秦拯. GARCH 模型在计算我国股市风险价值中的应用研究[J].系统工程理论与实践,2003,5:20-25.
[22] 张玉春.中国股市收益的 ARCH 模型与实证分析[J].首都经济贸易大学学报,2006：84-88.
[23] 孙星.时间序列 ARCH 模型在金融领域的研究[D].苏州大学,2013.
[24] 陈文伟.数据仓库与数据挖掘教程[M].北京：清华大学出版社,2006.
[25] http://www.cnblogs.com/phoenixzq/p/3539619.html.
[26] http://my.oschina.net/letiantian/blog/324269? p=1.
[27] http://cos.name/2013/01/drawing-map-in-r-era/.
[28] [美]Winston. R 数据可视化手册[M].肖楠,邓一硕,魏太云,译.北京：人民邮电出版社,2014.
[29] [印]Vignesh Prajapati. R 与 Hadoop 大数据分析实战[M].李明,王威扬,等译.北京：机械工业出版社,2014.
[30] http://zh.wikipedia.org/wiki/%E6%B3%95%E8%98%AD%E8%A5%BF%E6%96%AF%C2%B7%E9%AB%98%E7%88%BE%E9%A0%93.
[31] http://zh.wikipedia.org/wiki/%E5%8D%A1%E5%B0%94%C2%B7%E7%9A%AE%E5%B0%94%E9%80%8A.
[32] http://baike.baidu.com/link? url = vyo5N2qg3Z_gW90tzW9KSIyjBVK4Lmicd3G05iXEYD8hHek4zPfLLLathELYE7GnQYJDqIEjL-Ao5ElLt2GKBa.
[33] http://baike.baidu.com/link? url = abB8sdsDSaPQktqf1R5ErXwoceKpmjWUuAt_7dkzirxNecrJ

YALOJXiD5_ jWu39hW_XYOwNyG3G7krQLvWX0tK.
[34] http://wenku. baidu. com/link? url=QrbFPZLd0F1MR4RrFWwLr 8v5UQ4Ej8NqazqIC4L64aJ60p3Gnx1poEf0AYVZoVudN-ttz3Lp9fheRLClllXUftz8W-xbU0_Pb7dngukBTIu.
[35] http://jpkc. nwpu. edu. cn/jpxin/gccsjs/class/1/1. 3. 4. htm.
[36] John Maindonald. Data Analysis and Graphics UsingR[M],2008,71-74.
[37] http://baike. baidu. com/link? url=dvvacMegIr0y8yVM0tR0pg4yiYpbVaj8arV7OAlW9HfwQzdd5V5fw6NlRbBMXjfz6pu0552X88F-Z8tPwI6Bra.
[38] http://blog. csdn. net/ariessurfer/article/details/41310525.
[39] http://blog. sciencenet. cn/blog-984197-821456. html.
[40] http://www. cnblogs. com/wentingtu/archive/2012/03/03/2377969. html.
[41] 刘忠宝,王士同. 改进的线性判别分析算法[J]. 计算机应用,2011,31(1):250-253.
[42] 曹玲玲,潘建寿. 基于 Fisher 判别分析的贝叶斯分类器[J]. 计算机工程,2011,37(10):162-164.
[43] 崔自峰,吉小华. 基于线性判别分析的特征选择[J]. 计算机应用,2009,29(10):2781-2785.
[44] 刘靖明,韩丽川,侯立文. 基于粒子群的 K 均值聚类算法[J]. 系统工程理论与实践,2005(6):54-58.
[45] 朱明. 数据挖掘[M]. 合肥:中国科学技术大学出版社,2002:129-159.
[46] 谢娟英,蒋帅,王春霞,张琰,谢维信. 一种改进的全局K均值聚类算法[J]. 陕西师范大学学报(自然科学版),2010,38(2):18-22.
[47] 贾瑷玮. 基于划分的聚类算法研究综述[J]. 电子设计工程,2014,22(23):38-41.
[48] 任景彪. K-均值聚类算法的研究与分析[D]. 天津工业大学,2011.
[49] 李靖华,郭耀煌. 综合评价的多元分析方法——主成分分析法[J]. 管理工程学报,2002,16(1):39-43.
[50] 李树清. 改进的主成分分析法在综合评估中的应用[J]. 济宁学院学报,2010,3(31):15-17.
[51] 洪素珍. 如何有效利用主成分分析中的主成分[D]. 华中师范大学,2008.
[52] 高艳,于飞. 一种用于综合评价的主成分分析改进方法[J]. 西安文理学院学报:自然科学版,2011,14(1):105-108.
[53] 张鹏. 基于主成分分析的综合评价研究[D]. 南京理工大学,2004.
[54] 饶从军. 因子分析法中因子得分的岭估计[J]. 西南民族大学学报:自然科学版,2004,30(2):138-140.
[55] 邵晓锋,李龙星. 因子分析中因子得分的估计[J]. 黄冈职业技术学院学报,2003,5(4):81-84.
[56] 郭淑会. 因子得分的广义岭估计[J]. 孝感学院学报,2009,29(3):41-43.
[57] Sean Owen, Robin Anil, Ted Dunning, Ellen Friedman. Mahout in Action[M]. 2012,13-25.
[58] http://baike. baidu. com/link? url=BDYZOdVp35zzwOKJ99A-T84CjXvhcDY7BB8aCpIyBPAxf7VKL7LpxwOGHpdtMDSMjCra9tPXA8e14npMZrsJtq.
[59] http://blog. csdn. net/johnny710vip/article/details/23703931.
[60] http://wbj0110. iteye. com/blog/2068300.